

國立台灣大學工學院土木工程學系

碩士論文

Department of Civil Engineering

College of Engineering

National Taiwan University

Master Thesis

以 BIM 與代理人技術實現

營建協同設計審查之研究

Collaborative Design Review in Construction Using
BIM and Agent technology



何松柏

Ho, Soung-Bo

指導教授：謝尚賢教授

Advisor : Prof. Hsieh, Shang-Hsien

中華民國 99 年 6 月

June, 2010

國立臺灣大學碩士學位論文
口試委員會審定書

以 BIM 與代理者技術實現營建

協同設計審查之研究

Collaborative design review in construction

Using BIM Agent technology

本論文係何松柏君（R97521609）在國立臺灣大學土木工程學系
碩士班完成之碩士學位論文，於民國 99 年 6 月 21 日承下列考試委員
審查通過及口試及格，特此證明

口試委員：

謝尚賢

(簽名)

(指導教授)

廖源輔

周頤宇

庄仁仲

呂良正

(簽名)

系主任

誌謝

本論文能夠完成，首先必須感謝的是恩師謝尚賢老師，老師於兩年內用心指導與尊尊教誨，不管是學術或者生活做事上，都讓學生獲益良多，並且也提供了優質的研究環境與許多資源，讓我們在研究上不會因條件限制，而有所侷限。接著必須感謝的是中鼎工程顧問公司的廖源輔副理，廖副理於兩年間協助甚多，並提供許多實務上之建議，讓論文更加的實務與實用。也感謝康仕仲康老師及中興工程顧問公司研發及資訊部的周頌安經理，在口試期間提供寶貴建議與指導，使得論文更加充實與完整。

感謝 CAE 的大家庭成員們，大家一起互相砥礪鼓勵與打氣，常常讓陷入困難中的我重新振作起來，並且可以藉由互相討論，獲得更多的想法與建議，真的感謝 CAE 各位的陪伴。特別感謝君帆學長的大力幫助，在呈現疑惑時候即時給我建議，讓整個研究更加順利，還有宏霖學長在程式上以及技術上的幫忙，讓我能順利的完成研究之程式，還有其他的 CAE 夥伴：提供許多建議的乃文學長，還有雖然近來不到一年，卻很融入大家的奕竹，以及同屆做事熱心的凱禎、很有效率的千嘉、默默幫忙的正偉、總是充容不迫的育誠、很有規劃的謄兆、做事乾淨的亞欣、善解人意的依玲、非常厲害的煜璋，謝謝您們的陪伴，把苦悶的研究生活也點綴的多采多姿。

接著感謝土木系所許多朋友，結構組的裕堂、建成，由於他們陪伴讓我在學術之外，擁許多的空間可以放鬆心情，對於我緊張的研究生活有著非常重要的幫助，除此之外營管組的翊楷與學弟峻品，幫助我在論文寫作上提供許多意見，碩士生活謝謝您們的加入，有了您們我才不會只在學術上學習而已，還學習到許多學校沒教的事。

最後當然是我最親愛的家人們，永遠給我最大的鼓勵與幫助的姊姊，總是給我無微不至的照顧的媽媽，默默的教導我的爸爸，以及雖然不常見面的哥哥，您們讓我毫無後顧之憂的專心在研究上，謝謝您們。在此我將論文完成時的喜悅心情與您們分享。

何松柏 謹誌 於台大土木工程研究所
中華民國九十九年七月

摘要

科技隨著時代的改變與進步，現代建築物包含了各式各樣的系統，在機能方面比以往更加複雜。除了基本的機電與給排水系統(Mechanical , Electric , Plumbing 簡稱 MEP)、空調、弱電和通訊外，消防保全、節能設備、光纖網路與智慧型建築(Intelligent Building)的各種感應器和控制系統，也逐漸成為現今建築的標準配備。隨著建築物構造複雜度增加與所包含的系統越多，協力廠商在設計與施工階段的整合需求也大幅增加。不同系統(例如：機電與排水系統)的設計模型，必須在施工前進行整合與審查其介面相容性。多樣性的系統整合需倚賴多位專門業者的協調分工。但因各業者多分散異地而增加了見面討論的困難度，所以現今多需要以電話聯絡做為溝通管道。然而，由口語來進行 3D 空間概念的描述常不易理解；因此，在溝通上常常耗費過多的人力與時間。

本研究提出一個介面平台連接網路與 3D 模型展示的應用程式。引入物件導向程式概念與開放代理者架構(Open Agent Architecture)，實作 BIM Review Agent(以下簡稱 BIM.RA)。審查者可透過使用 BIM.RA 經由網路於異地進行協同設計審查。利用 3D 模型的展示，協助表達圖面與文字描述難以表達的 3D 資訊也是 BIM.RA 的重要特色。且針對複雜的建築系統，BIM.RA 可以有效率的幫助審查者在 BIM 模型中找到對應的視覺化物件，以利快速解決系統介面衝突(例如：空間衝突、管線互相干擾...等)方面的工程問題。

關鍵詞:代理者、協同審查設計審查、建築資訊模型

ABSTRACT

As the technology improved with time passing by, the modern building is equipped with more systems. In addition to the basic systems, like MEP (Mechanical, Electric, Plumbing), drainage system, water supply, air-conditioning (AC), electronic and telecommunication system, the building usually possesses more complex systems to be modernized. For example, the fire fighting system, the energy-saving equipment and the security network are frequently equipped today.

In the design phase of a project, each team constructs its own 3D model for the specific system. Before executing the design, all team's built models need to be carefully inspected to be integrated. Because conflicts might exist among different systems' designs, design teams would spend a lot of time to discuss and resolve them. The project with more complex systems requires more frequent discussions. When teams are distributed in different places, communication via the telephone is conventional. However, 3D concepts are hard to be interpreted by mere words. This makes it difficult for participants to comprehend the discussion issues. Hence, this high degree of the co-ordination causes time and money consuming. Hence, to cause the waste of time and money in communication.

In this study, we propose an Internet-accessible interface which integrates 3D display applications. We complied with the open-agent architecture to implement an object-oriented prototype system, call BIM Review Agent (BIM.RA). Inspectors are able to use BIM.RA to coordinate from anywhere with Internet. It is important that BIM.RA applies BIM model to express 3D information of objects which is too complicate to describe by graphics and characters. In this way, BIM.RA allowed users to explore the visualized objects efficiently and to resolve the conflicts faster among different system models (ex interference check and clash check).

Keyword: Agent, Collaborative design review, BIM

論文目錄

第一章 緒論	1
1.1. 研究背景與動機	1
1.2. 研究目的	3
1.3. 研究方法與步驟	4
1.4. 論文架構	6
第二章 協同設計審查與 BIM	9
2.1. 協同設計審查	9
2.1.1 協同作業簡介	9
2.1.2 協同設計介紹與分類	12
2.1.3 協同設計審查定義	15
2.2. 建築資訊模型(BIM)	20
2.2.1 建築生命週期簡介	20
2.2.2 建築資訊模型定義	23
2.2.3 CAD 與 BIM 技術發展情況	24
2.3. BIM 技術之協同設計審查應用	26
第三章 代理人機制架構設計	29
3.1. 需求分析	29
3.2. 代理人設計	31
3.2.1 開放代理人架構(Open Agent Architecture)簡介	31
3.2.2 BIM 代理人應用	32

3.3. API 設計.....	34
3.3.1 基本 API 設計	35
3.3.2 高階 API 設計	35
3.3.3 Server 與 BIM.CS API 設計	37
3.4. 本章小結	38
第四章 BIM.RA 與 BIM.CS 的設計與實作.....	39
4.1. 設計概念與分析	39
4.1.1 設計難題	39
4.1.2 比較分析	41
4.2. BIM 協同操作之種類.....	42
4.2.1 精簡型用戶端(Thin client)架構定義.....	42
4.2.2 完整型用戶端(Fat client)架構定義.....	44
4.2.3 複合式用戶端架構定義.....	45
4.3. 物件導向設計	48
4.3.1 物件導向技術概念.....	48
4.3.2 物件導向設計模式.....	49
4.3.3 擴充性	50
4.3.4 系統使用者	50
4.4. 軟硬體架構	51
4.4.1 程式語言	51
4.4.2 可延伸標示語言(eXtensible Markup Language).....	52
4.4.3 繪圖套件 SmartPlant Reviews(SPR)介紹	53

4.4.4 程式架構	53
4.4.5 BIM.RA 與 BIM.CS 使用與運作流程	54
第五章 電腦輔助設計審查展示案例	57
5.1. 設計審查案例說明	57
5.2. 模型資料	58
5.3. 執行流程與結果展示	61
5.3.1 操作流程概述	61
5.3.2 結果展示設定	63
5.3.3 操作情境	65
第六章 結論與建議	69
6.1. 結論	69
6.2. 未來研究建議	72
參考文獻	73
附錄 A: BIM.RA 基本 API	77
附錄 B: BIM.RA 高階 API	79
附錄 C: Server 與 BIM.CS API	83

表目錄

表 2-1 生命週期各項費用分布	22
表 2-2 BIM 管理工具條件	27
表 3-1 生命週期溝通不良損失(單位:百萬)	30
表 3-2 華建資訊分類	30
表 3-3 基本 API 分類	35
表 3-4 高階 API 分類	36
表 3-5 BIM.CS API	38
表 5-1 案例說明表	57
表 5-2 衝突點說明表	59
表 5-3 衝突解決順序表	59
表 5-4 案例衝突點對比顏色比較	64



圖目錄

圖 1-1 研究流程圖	5
圖 1-2 研究架構圖	7
圖 2-1 協同分類	11
圖 2-2 協同設計分類	14
圖 2-3 建築生命週期中品質與成本影響程度	15
圖 2-4 協同設計審查分類	19
圖 2-5 建築生命週期	21
圖 3-1 代理人分類	31
圖 3-2 開放代理人架構	32
圖 3-3 BIM.RA 開放代理人	33
圖 3-4 Server XML 檔案	37
圖 4-1 Thin Client(I)	43
圖 4-2 Thin Client(II)	43
圖 4-3 Fat Client	45
圖 4-4 複合用戶端	46
圖 4-5 設計模式架構	50
圖 4-6 XML 檔案	52
圖 4-7 JNI 程式架構	54
圖 4-8 伺服器端作業流程	55
圖 4-9 混合端作業流程	55
圖 5-1 案例情境	58
圖 5-2 案例模型	60
圖 5-3 案例衝突點	60
圖 5-4 與會者 XML 登入畫面	61

圖 5-6 協同流程	62
圖 5-7 案例模型對比顏色	63
圖 5-8 流程情境	65
圖 5-9 審查流程情境	66
圖 5-10 同步畫面	67
圖 5-11 經過使用者自行調整過的畫面	68



第一章 緒論

1.1. 研究背景與動機

隨著時代變遷、科技進步，建築物於機能方面較以往更加複雜，其中包含各式各樣之系統，除基本機電與給排水系統(Mechanical , Electric , Plumbing ;MEP)、空調、弱電與通訊外，消防保全、節能設備、光纖網路及智慧型建築(Intelligent Building)研究裡面之各種感應器與控制系統，成為現今建築之標準配備。

然而建築物構造越複雜、系統越多，會導致設計與施工時期產生協力廠商整合問題；不同系統(如機電、排水等)所設計之模型，將於施工前整合與審查，許多介面問題需在此階段解決。因各業者分處不同地方，見面討論有一定之難度，現今常藉由電話溝通交換意見與想法，並且也可能因不可避免之因素，不能見面開會例如 SARS 與 H1N1 流行時間，然而口語描述不易理解；口述者必須具有良好 3D 概念，聆聽者也必須想像或猜測 3D 空間位置，故常耗費人力與時間在溝通上。建築資訊模型(Building information model;BIM)(Eastman et al., 2008)為近年來營建業新興之概念，BIM 是把建築生命周期(設計、建造、使用到拆除)中完整資訊置於 BIM 模型，目前研究指出 BIM 能有效地整合建築系統資訊與節省時間及成本。

設計審查之方式，雖然已開始使用 BIM 模型審查，審查完畢後仍多採用傳統 2D 圖說與資料文件進行保存施工完成後相關資訊。但 2D 圖說欲表現出立體概念，則依憑製圖者的繪製技巧良好，觀看者之空間概念良好，雙方才可進行概念傳遞，故使用 2D 圖說所產生缺點為：複雜界面不易理解、檢視圖面不易、無法以簡明扼要表現方式呈現建築物現況與資料龐大且複雜需花費大量時間搜尋需要的資訊，因此讓

審查人員審查時容易產生錯誤與花費多餘時間，甚至仍可能找不到資料來源（吳崇弘，1997）。

隨著電腦輔助技術進步、硬體運算能力之強化與網路技術之提升，透過網際網路數位傳輸與 BIM 軟體顯示 3D 建物模型之能力，專業分工異地協同設計，讓身處不同空間之設計人員，能快速的藉由 3D 模型了解討論之議題，大幅提升設計審查爭議解決效率，希望透過網際網路傳輸資訊，利用代理人架構(參考 3.2.1)為基礎，建立一個程式架構（The Open Agent Architecture, 2010），提供身處於架構的與會人員，包括訊息傳遞與視覺化展現之服務。於代理人架構下，參與審查之工作人員能夠互相觀察、溝通與互動。藉由整合網路與 BIM 軟體發展出一個具溝通能力之方法，是本研究探討之重點。



1.2. 研究目的

本研究為應用代理者技術與 BIM 軟體應用，以建立網路協同審查作業之架構，提供異地協同審查人員，同步 3D 畫面與即時資訊，並藉由 3D 協同操作與 BIM 資料整合，支援審查人員或不同合作廠商，進行設計審查，以解決空間衝突與整合工程介面。

為了達成研究目的，首先探討複雜建築系統其難以藉由圖面、文字表達描述之 3D 資訊、多系統介面不易理解等問題，比較目前電腦使用端，提出一個具有網路連接功能之程式架構。接著討論設計該架構時，須考慮技術門檻與實作成本，並藉由物件導向概念與設計模式的方法，開發一名為 BIM Review Agent (以下簡稱 BIM.RA) 程式介面工具，來達展示 3D 模型之效果。

BIM.RA 工具開發之概念包括：為了解決異質軟體間之協同作業，所設計之代理者架構；與發展新的使用者用戶端功能，以解決傳統 3D 模型軟體，不具有網路操作之間題。利用 BIM.RA 可達到藉網路即時呈現 3D 的模擬與視覺化效果，幫助協同審查人員即時同步 3D 畫面資訊，再透過 3D 協同操作與 BIM 資料整合，支援審查人員或不同合作廠商，進行空間衝突解決與工程介面整合，並且應用在協同設計審查上。

1.3. 研究方法與步驟

基於研究預計達成之目標，將研究步驟流程如圖 1-1。主要分為三大階段：文獻收集階段、資料分析與技術評估階段與架構提出與實作探討階段。

確定動機與目標之後，首先透過相關文獻回顧，了解本研究相關之議題發展以及目前應用。本研究針對之議題如下：首先對於協同設計審查作探討，此議題分為協同與設計審查；於此先對協同的條件及其相關研究進行回顧動作；其次對於設計審查定義作討論，於是整理協同設計綜合三個區塊之文獻(協同、設計審查與協同設計審查)；接著定義協同設計審查，了解協同設計審查目標，並選定工具，就其發展性、擴充性與方便性討論。

本研究所選定工具為 BIM 模型與開放代理者架構(參考 3.2.1)，並協助系統架構開發；BIM 為營建業所新興之技術，BIM 具備較完整的建築生命週期之資訊，並且已有許多學者利用其改善營建品質與成本問題，而 OAA 架構推出將許多不同性質軟體在一個分散之環境(異地，如網際網路)下例如：網際網路，組織協同工作。

藉由上述結果，進行 BIM 協作審查機制之設計與分析，並把程式架構分成兩個獨立的架構，並針對市面 BIM 相關軟體利用 OAA 的架構，提出一個複合式使用者端架構，讓其能獨立運作，提供平台給使用者開發相關應用，再設計一個協同審查核心程式，架構在上一部分的平台上，就此開發了名為 BIM.RA (BIM Review Agent)平台與 BIM.CS(BIM Coordination Service)兩個程式。因本研究探討 BIM 軟體配合網路技術應用在協作審查，為因應各家不同 BIM 軟體，於是提出一個適合不同 BIM 軟體並針對其實作之架構，由於架構並不限定 BIM 軟體，因此本研究選擇 SmartPlant Review(SPR)，進行實作開發，

對於架構而言只要符合條件(參考 4.3.3 節)的 BIM 軟體之語言，皆可根據此架構進行開發。

提出架構與實作階段，因驗證程式架構與初步實作較困難，所以採用雛形開發模式。先開發出系統雛型，其不斷修正架構與程式功能，以期符合使用者需求。最後利用實作結果探討 BIM 軟體和視覺化軟體與網路之互動與整合模式，以及未來發展方向，作為研究結論。

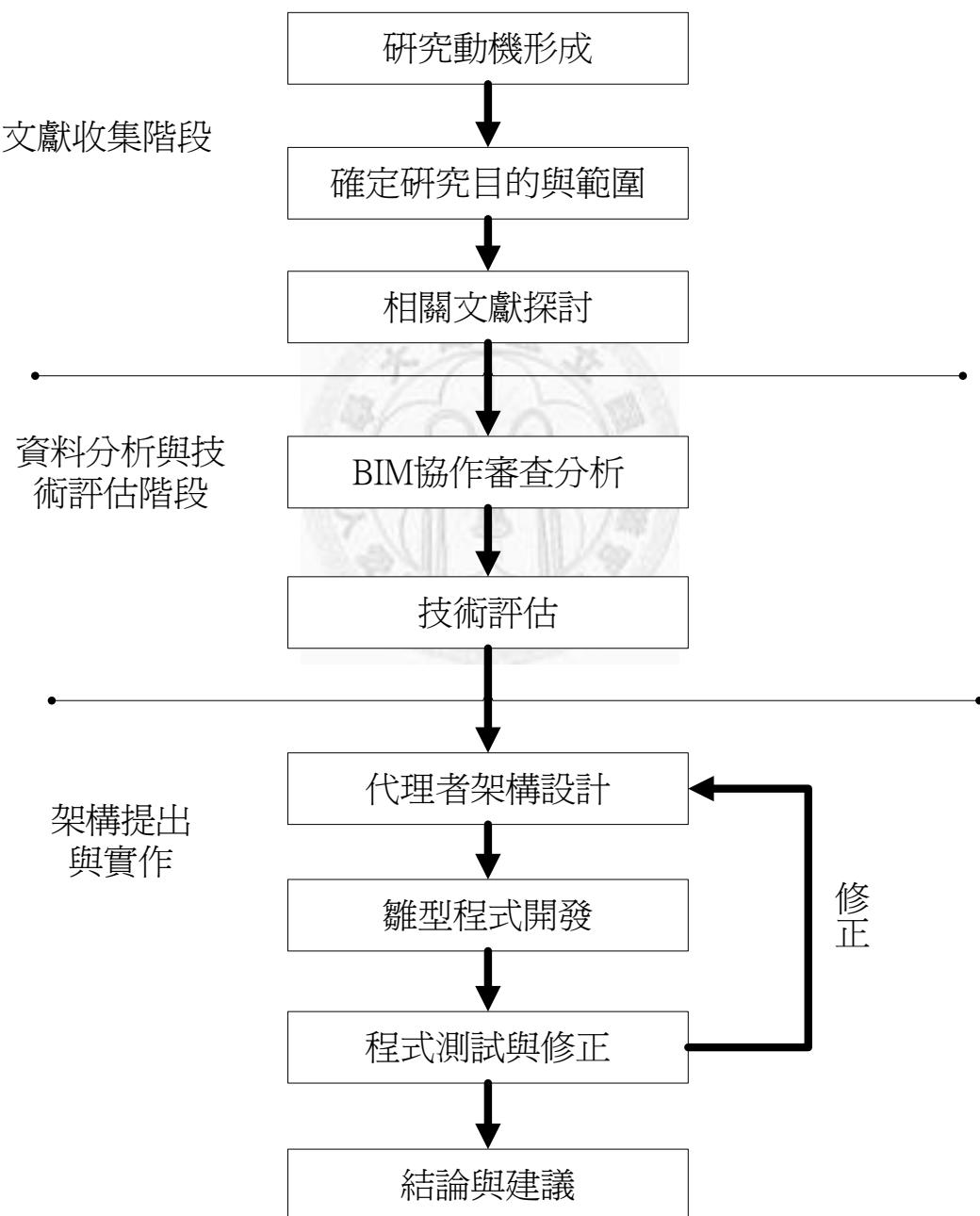


圖 1-1 研究流程圖

1.4. 論文架構

本研究之論文架構，依據研究架構如圖 1-2 所示，整理成下面章節，在論文中之紀錄是先以應用之範圍與目標作說明，接著討論技術與設計概念，然後為應用案例呈現，最後為結論與建議，以下將逐一說明各章之重點。

第二章 協同設計審查與 BIM

本章將定義以下名詞：協同設計審查與建築資訊模型作定義，並分別說明其演進過程，且陳述協同審查的問題之所在。

第三章 介紹代理者機制架構設計

本章承續第二章，就問題深入分析，然後介紹代理者機制，利用此機制開發一個能分析優缺點與適用情境的架構，應用於設計審查上。

第四章 BIM.RA 與 BIM.CS 設計與實作

此章節利用第三章所提出之架構與複合式使用端，實作 BIM.RA 之離形程式，分別以物件導向概念與軟硬體架構，探討限制與擴充性，最後並介紹 BIM.RA 使用流程。

第五章 電腦輔助設計審查展示案例

應用 BIM.RA 級形程式於設計審查案例上：首先介紹案例的情境，並且對使用程序加以說明，針對資料需求作探討，最後展示應用成果後分析。

第六章 結論與建議

針對本研究實作與分析結果做出結論，檢討系統所需改進之處與提出未來研究方展方向。

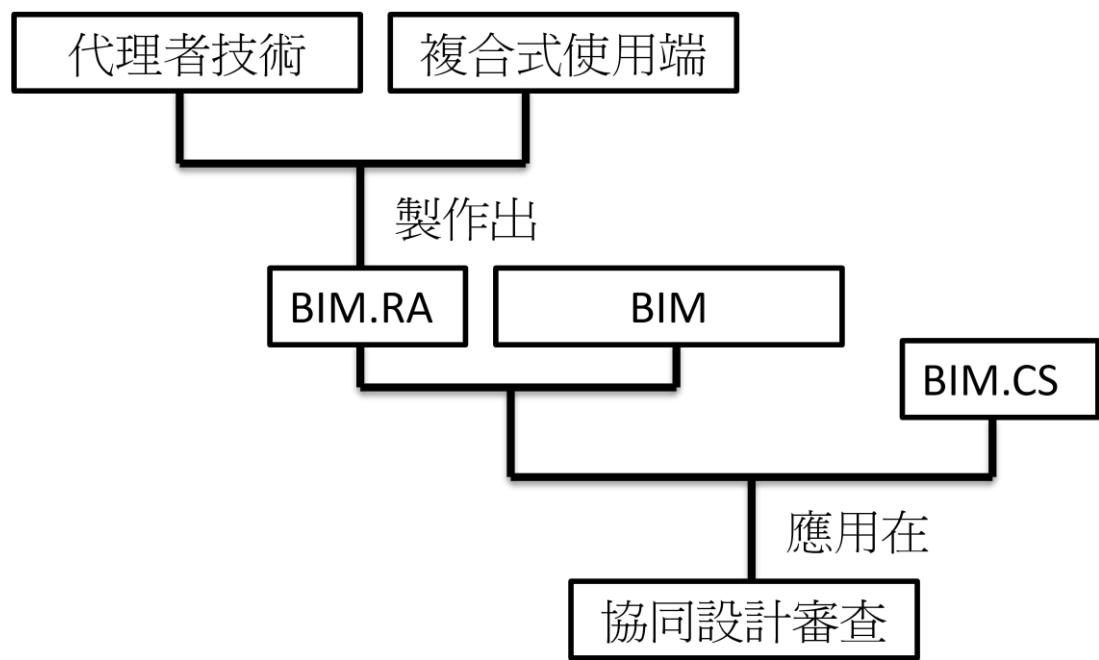


圖 1-2 研究架構圖



第二章 協同設計審查與 BIM

本章針對以下名詞作定義：建築資訊模型與協同設計審查；分別交代其演進過程，最後陳述協作審查之問題所在。

2.1. 協同設計審查

本研究所描述之協同設計審查概念，來自於協同設計，此兩種概念(協同設計審查與協同設計)皆來自於協同工作，由於電腦之進步，許多傳統作業模式面臨著改變。尤其是營建產業，因產業歷史悠久，早在電腦普及前，協同作業就已被需求，但當時的時空背景與現代大不同，作法應有所改變，以下將就協同作業、協同設計與協同設計審查逐一介紹。

2.1.1 協同作業簡介

協同作業早在 1999 年時，由 Fruchter (1999)提出協同依時間分成同步協作(Synchronous collaboration)與非同步協作(Asynchronous collaboration)，然後接著再以方法分類，空間的方法與共享內容的方法區分，第一種方式指的是真實空間與虛擬空間皆算，但是討論中沒有分享資料，真實空間是同處於一地，虛擬空間則是協同人員留在自己的辦公室，利用網路達到討論的效果(如視訊會議)，第二種指出，協同成員分享資料達到協同效果，而 2002 年 Anumba et al. (2002)更進一步整理協同形式，依據時間與空間分成四類，如圖 2-1 協同分類 (A)所示。

- 面對面之協作方式為最普遍的協作方式；協作人員同時、同地(如會議室)就專案進行討論，如營建業之審查，聚集各方面人員就一項建築專案在同一地做討論；

- 不同時但是同地之協作方式，常利用記事板或公告板，將欲傳達訊息記錄其上，讓其他人可以藉由公告板或記事板，了解各人所提出意見並達成協同作業；
- 同時但分散(不同地)之協作方式，利用現代之技術如電話或網際網路，打破原本因地理環境因素而分開之界線；
- 不同時且分散(不同地)之協同方式，這種協同方式將訊息或資訊，經由傳遞而傳達給他人，收訊息的人不一定要在特定的地點才可以接收訊息，例如：電話留言、電子信箱與紙張傳遞。

本研究只針對於分散(不同地)的協同做探討，因同地也是分散中一種狀況，故若是有一種方式可用於分散的狀況，也能適用於同地之情況，故本研究將協同矩陣簡化成為圖 2-1(B)，並且對於不同地做出更嚴苛的定義，本研究所指出地不同地，為人員處於不能直接交談(不透過工具)之地理位置，例如公司一樓與二樓或台灣與美國。



	同時	不同時
同地	面對面協同方式	不同時的協同方式
不同地	同時但分散的協同方式	不同時且分散的協同方式

(A)協同分類矩陣(Anumba et al., 2002)

	同時	不同時
不同地	<div style="border: 1px dashed black; padding: 2px;">面對面協同方式 (同地)</div> 同時但分散的協同方式	<div style="border: 1px dashed black; padding: 2px;">不同時的協同方式 (同地)</div> 不同時且分散的協同方式

(B)協同分類整合

圖 2-1 協同分類

於協作之技術上可分成兩大類 (Shen et al., 2010)，第一類為以網路為基礎之資料分享，將建築生命週期資訊利用網際網路置放一個地方，需要資訊則上網去獲得資料，且可將資料更新，協作人員通常只對同一資料做處理，不會有資料版本錯誤之問題，屬不同時且分散之協同方式；另一種以代理人為基礎之技術，在網路還未出現前，軟體代理人之概念早已被廣泛應用，代理人常被應用於使不同性質的軟體，互相合作與協同作業的功能上，再加上網際網路的發展，需互相協同作業之程式已經不必要在同一台電腦上，可用網路串連不同代理人分工合作，例如 OAA 架構(參考 3.2.1 節)。

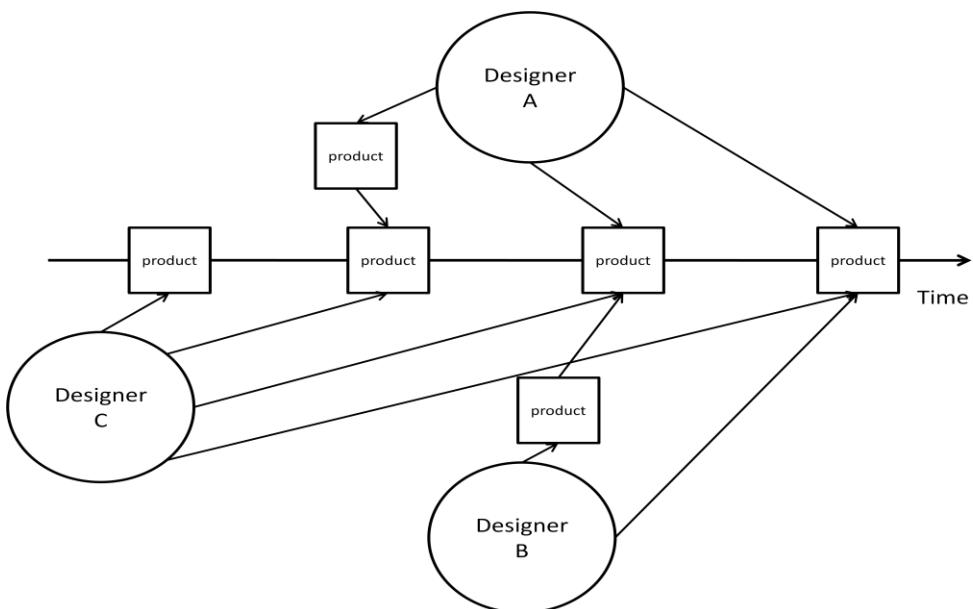
2.1.2 協同設計介紹與分類

3D 設計之發展主要是為改善繁瑣、費時與精度低之傳統手工繪圖，隨 CAD 與 BIM 之發展(參考 2.2.3)，雖然改善傳統繪圖之設計方式，但仍存在許多缺點；設計工作依然只能獨力完成，當面對複雜之設計工作必須耗費相當大量時間與他人溝通，即使將龐大模型拆解成許多小模型來進行設計，但依然需耗上相當多時間與精力進行小模型之間之界面整合。

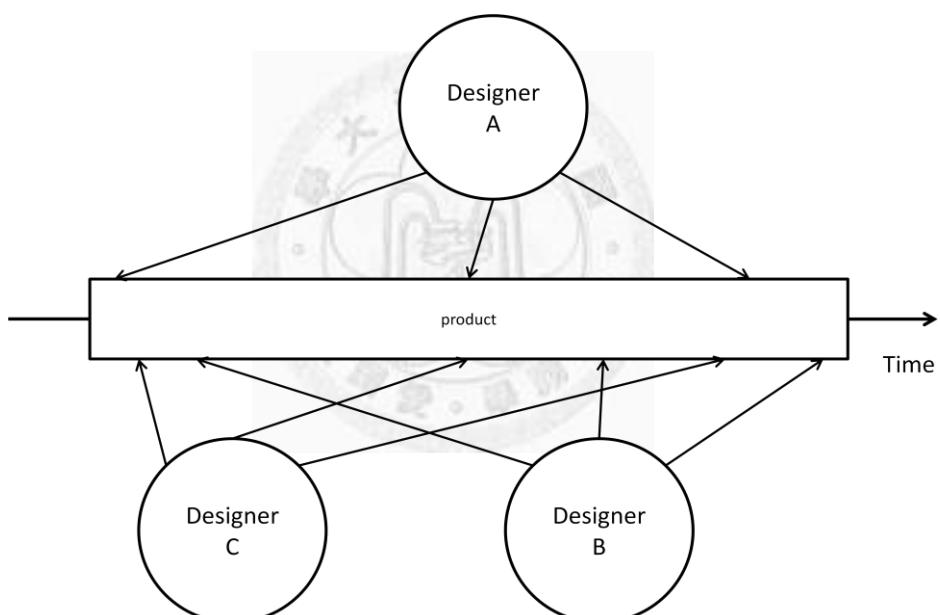
由於網路的發展，協同設計之技術也不斷被開發，依照技術區分成兩類 (Shen et al., 2010)，第一類為網路為基礎之協同模式 (Web-based collaboration)，此模式主要是設定眾人皆認可之格式與規則，資料須經由網路更新來保持最新版本，團隊成員修改設計後將資料利用網路傳至伺服器；第二類為代理人之協同模式(Agent-based collaboration)，代理人所扮演角色，通常用於促進合作與協調軟體之功能，用於加強與溝通於協作軟體、系統用戶、組織與硬體間；利用以上兩種模式，可虛擬協同環境(Collaborative virtual environments)，隨虛擬技術、代理人技術與網路技術逐漸廣泛使用下形成許多架構，而設計者在這些架構下，與其他成員於虛擬的空間一起設計；也可以虛擬一個中間協調者(Virtual organization as a collaboration medium)，利用其組織，達到資料交換與合作之目的。A.N. Baldwin 等人提出，代理人機制之協同方式非常適合應用於同時但是不同地之協同方式。

協同設計可分成兩種型式(Kvan, 2000)，第一種協同設計方式如圖 2-2(A)所示，其中參與者有 Designer A(以下簡稱 A)、Designer B (以下簡稱 B)與 Designer C(以下簡稱 C)。A、B 與 C 三人於不同地方，

設計相同案子，一開始以 C 將設計之檔案作為基礎，之後 A 將設計之檔案，加入以 C 為基礎檔案中，成為 C 與 A 之協同檔案。然後 B 也同樣將設計好之檔案，置放 C 與 A 合作之檔案中，成為最後三人協同設計之檔案，此方法於在協同分類中不同地方與不同時間之方式，最重要之處為參與協同的人員，規定個別設計之檔案格式須相同，設計分工要精確，於營建業上此類的協同設計研究，有許多方式可達成，目前最常見之方式為利用 IFC(Industry Foundation Class)，IFC 是由 IAI(International Alliance Interoperability)組織——許多繪圖軟體公司與學術界，針對營建工程提出一套專屬於營建工程資訊模型的標準，利用此標準 Chen et al. (2005) 將 IFC 資料儲存於網路上伺服器，包括設計、建築與結構三部分，再利用 JAVA 網路呈現技術，達到協同之效果，此外 Owolabi (2006) 也利用顯示或修改 IFC 標準所定義之物件，達成協同作業，Fruthter (1996) 利用分享 3D 模型，讓設計人員把設計概念與建議與模型間做連結，讓大家可以依據記錄來進行討論，還有 Chen (2007) 利用 JAVA3D 與 P2P(Peer-to-Peer) 之技術，將設計者資料分次傳輸，達到協同資料更新之目的，以上方式屬於 Web-base，第二種協同設計方式如圖 2-2(B) 所示，A、B 與 C 三人為各自在不同地方，但是他們設計之時間是同時，不需個別獨自作業時間，Abdel-Wahab (1988) 利用 Agent(代理人) 方式與網際網路，達到協同之效果，此種協同方式必須要代理人當中介，並且傳輸指令，故代理人跟代理人間必須有固定之溝通方法，例如 OAA 架構採用了自定義之代理人通信語言(Interagent Communication Language，ICL)，藉由這種方式指定特定之代理人工作，而本研究也參考 OAA 架構，但是代理人溝通的語言，採用本身定義之共同 API(參考 3.3)，代理人間溝通採取指令式。



(A)協同設計(不同時不同地)



(B)協同設計(同時不同地)

圖 2-2 協同設計分類(Kvan, 2000)

2.1.3 協同設計審查定義

工程前期規劃設計階段對於之後之作業成本有者重大的影響，如圖 2-3 所示，設計階段之作業為將需求與概念轉化成模型或圖樣，並且提出所需數量，因此設計階段最重要之目的即為，把設計需求與設計理念轉換成模型或圖面化，且轉換成設計圖與數量表，讓後續之工作能藉由設計成果計畫各階段之作業。設計成果實際展現了建構物之形式，並且牽動後續之作業，因此如何保證設計成果之資訊足以提供後續之作業，以及成功表達設計需求與理念，為設計審查之功能。

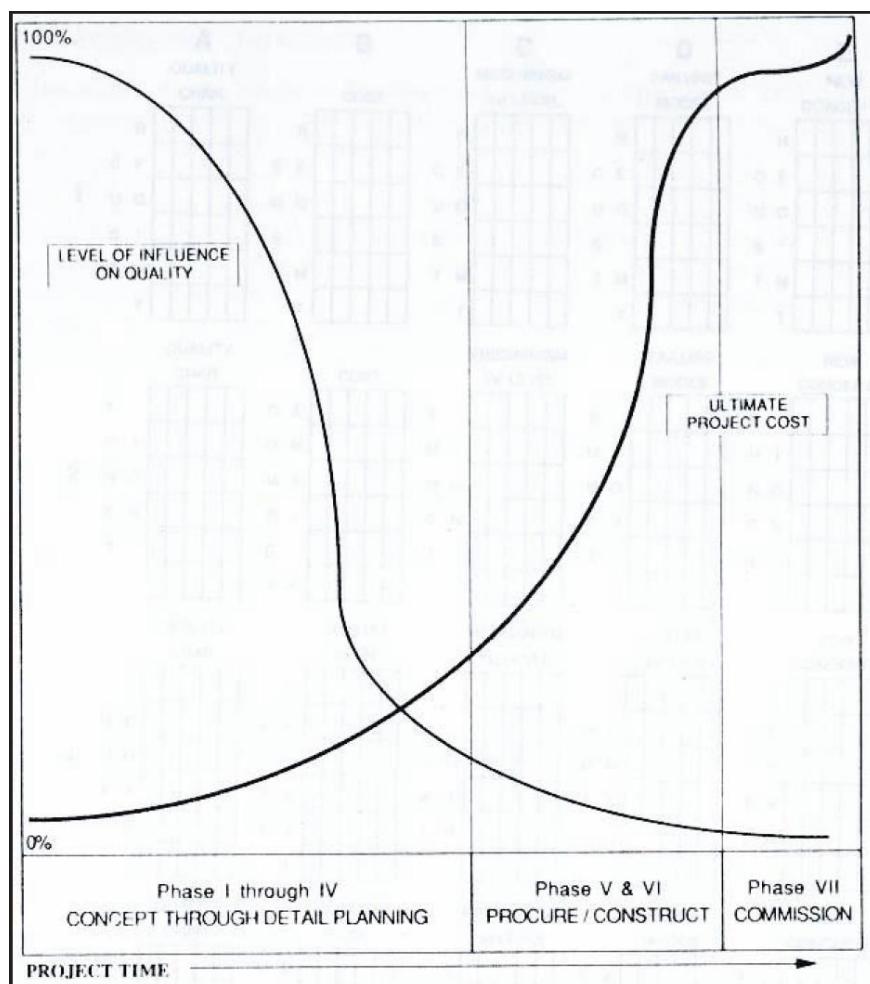


圖 2-3 建築生命週期中品質與成本影響程度

(Mallon and Mulligan, 1993)

設計審查常被當作一種手段用來提升技術品質並確保品質(李勝朗，1998)，於 ISO9001(2000 年)第七章指出，設計審查階段須評估設計與開發結果是否符合要求，並鑑別出所有問題且提出必要之措施，參與設計與開發之審查人員，應包括各職責部門代表，並應保留審查紀錄；市田嵩-牧野鐵治認為，設計審查為具體實現產品設計品質，對於計畫之產品、運輸、安裝、使用與維護之過程，需客觀地集中進行評價，並且建議改善，已推進至下一階段的一種組織之活動體系(市田嵩，1998)

設計審查之本質並非對設計者進行批評或對設計產品提出缺點，而是為在原有良好設計之基礎上，配合各方專家或有經驗人士協助設計者，預防失敗或造成成本偏高，以及事前對產品設計之分析改良等，進行一連串協助評估工作 (卓正倫, 2005)。設計審查被重視之理由，除產品之設計品質可被具體評量之外，因建築工程趨向複雜化，多機能化及各種系統變得複雜，在設計階段若能整合，將節省成本花費。

張清靠 (張清靠，1983)提出機械工業設計審查階段，而建築業也可運用其相關階段：

- 概念設計審查:亦稱系統設計審查(System Design Review)，依據建造業主或主設計工程師所提出開發計畫，如整體架構之概念與費用等需求、各階段時程之分配等設計構想進行審查，評估專案的最適性、相關性、完全性與風險。
- 初步設計審查:亦稱基本設計審查(Preliminary Design Review)，審查時機於未正式執行細部設計分析及測試之前，由概念階段所規劃之整體施工，進行整體專案之模擬分析及其助要工作流程，動態模擬分析結果進行分項評估，並確認設計所涵蓋之概念是否滿足需求。

- 細部設計審查:亦稱詳細設計審查(Critical Design Review)，從設計顧問公司所完成之設計藍圖與分析資料，最可能成為供應商資料，評估設計是否達成預定功能之可靠預測以期設計者獲得初步之肯定與信心、準備工程計畫與工程設計。

隨著 BIM 與協同設計之興起，設計審查應能具有協同審查之部分。利用 BIM 特性，可達到資料共享，幫助審查人員能夠打破空間與時間之限制，可在不同的地區，利用網路與協同廠商溝通，並可隨時進行，使工作能更緊密與彈性，進而減少所需之成本與時間。

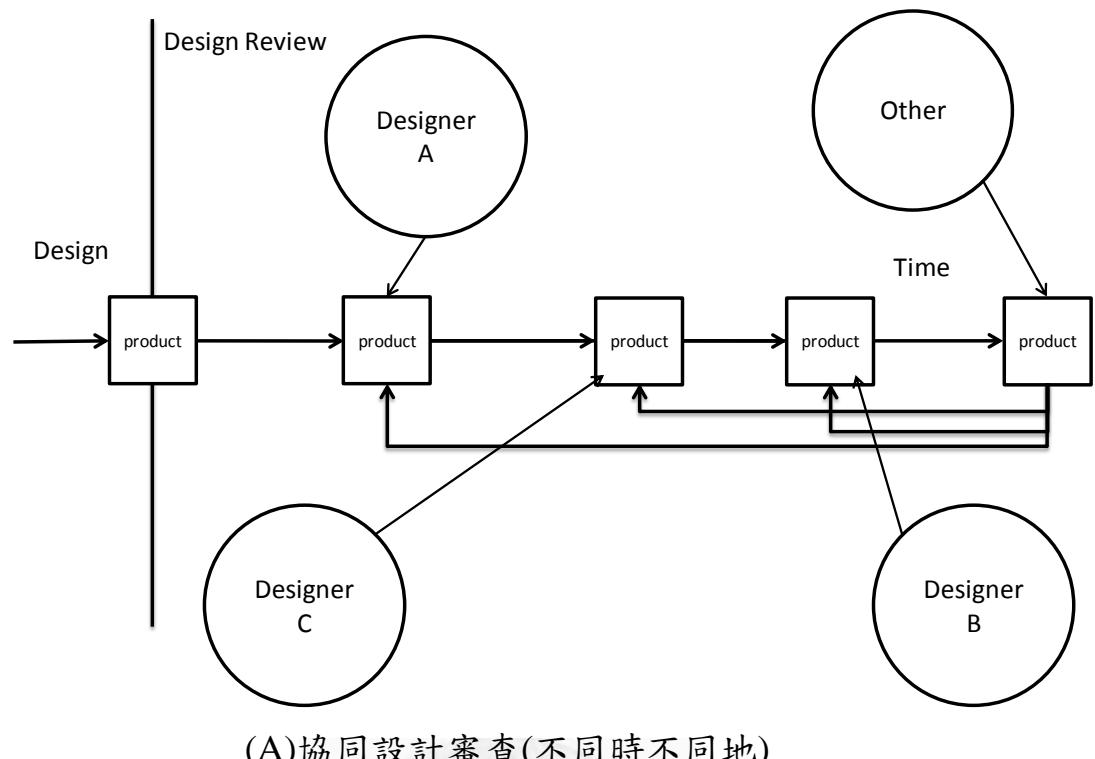
根據前兩節與設計審查之定義，可了解協同設計審查與協同設計息息相關，協同設計審查屬協同設計一部分，但由於審查者關注為設計的產品適應性、相關性、風險與是否滿足需求，而不是創造產品，故相對於協同設計所需考慮之因素較少，並著重於多種設計(給排水與機電等等)之介面溝通，模型的點線面或材質定位與選定，為電腦協助審查工具必須審慎考慮之重點。

依據協同設計分類為基礎，本研究將協同設計審查分成兩類，第一類依據前一小節圖 2-2(A)延續下來如圖 2-4(A)所示，設計完成之成果需其他人進行評估，依據各專案狀況之不同選定不同人參與，於示意圖中以”other”作為表示，譬如監造單位、業主機關等，當 other 審查之後發現與預期狀況不同或不符使用性，會將設計成果退回至負責設計者之手上，並且與原設計者做溝通，當設計者完成修改後，會通知參與設計團隊之其他設計者修改部分，團隊中其他設計者依據各自所設計部分進行探討，是否有不合適之變化，當相關設計者完成審查後，Other 會再進行審查，此種審查過程為緩慢之循環。另一種審查模式為接續前一小節圖 2-2(B)流程如圖 2-4 (B)所示，Other 與設計團

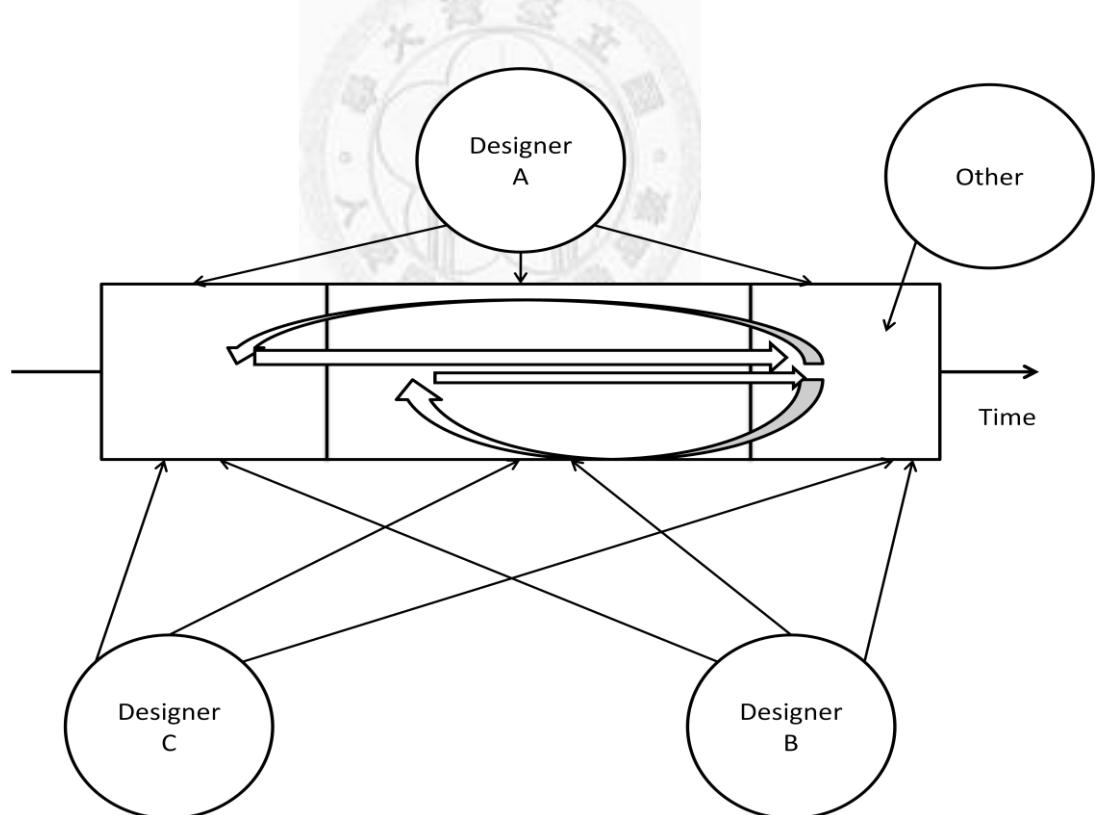
隊，於同時間一起藉由工具來幫助溝通，當其中一位設計者對於其他位設計者即時提出疑問，並且可立即獲得解答，並幫助協調設計與概念溝通，最後當 Other 就專案之規範作審視，若有問題可尋求支援，並且要求展示或澄清。

在系統設計上面，兩種模式所需要之特性並不相同，不同時且不同地的方式進行審查，重要的是審查之後意見與建議是否能被記錄，並且於下個審查人員能，依據紀錄了解之前審查人員之建議，同時不同地，則必須能有即時的把訊息傳遞給與會者，讓參與的人員能夠同步畫面，並且根據畫面進行討論，本研究將針對不同地但同時的審查方式與設計程式架構，進行實作架構與驗證可行性。

協同設計與協同審查在於軟體上需求不同，協同設計所用之軟體，重視的是大家如何一起建製模型，當然其中包含如何把協同成員創造資訊，傳遞給所有協同人員，如何保持檔案一致與最新，然而協同審查軟體則重視在於，大家如何看模型，也就是在審查階段已不能更改模型，而是針對此模型進行需求確認，因此其主要功能為如何讓大家都可以看到相同之物件。



(A)協同設計審查(不同時不同地)



(B)協同設計審查(同時不同地)

圖 2-4 協同設計審查分類

2.2. 建築資訊模型(BIM)

近年來建築資訊模型(Building Information Modeling ; BIM)興盛，軟體廠商跟進相關學會亦紛紛成立，其基本概念在於資訊不只可應用於設計施工階段，亦可應用於建築物完整生命週期，不只可建立創新資訊技術與商務結構，並大幅減少各種重複作業浪費與提升建築行業之低效能，以下各小節將會對建築生命週期、建築資訊模型和目前建築資訊模型的軟體做介紹，並且依據 BIM 特性將其運用在設計審查上。

2.2.1 建築生命週期簡介

生命週期為整個投資、使用與分析之時間，亦即一個計畫由開始至結束整段時間。故建築物之生命週期(Building Life Cycle)按照字面上意義，指建築物由構想、規劃、設計、施工、營運維護至翻修或拆除報廢等階段，所形成之一種過程，如圖 2-5 所示；於不同階段有不同之花費成本，會產生不同維護管理費用，故維護管理應針對欲管理之設施，採用生命週期概念。若使用建築生命週期來管理，則會提到生命週期成本，生命周期成本為建築物在生命週期間一切所發生之成本總和。在日本建築生命週期中各項支出項目之說明及費用分布如表 2-1 (蕭樂同，2004)；陳瑞鈴（陳瑞鈴等，2001）等由四個角度來界定建築物生命週期：

- 物理耐用年限（結構安全之生命週期）：因長期使用導致建築物結構老化，需拆除重建的建築壽命。
- 機能耐用年限（空間設備機能之生命週期）：因活動空間、工作效率、舒適健康上之機能老化與不堪使用，而決定拆除之建築壽命。

- 社會耐用年限（社會機能之生命週期）：因都市計畫變更、交通變遷、居住人口變化、環境惡化、地價上漲、停車空間不足等社會因素，而需提前拆除重建之建築壽命。
- 稅法耐用年限（財稅法之生命週期）：於在財稅法上為資產估價、銀行貸款、減稅償還等之估計，而規定之建築壽命。

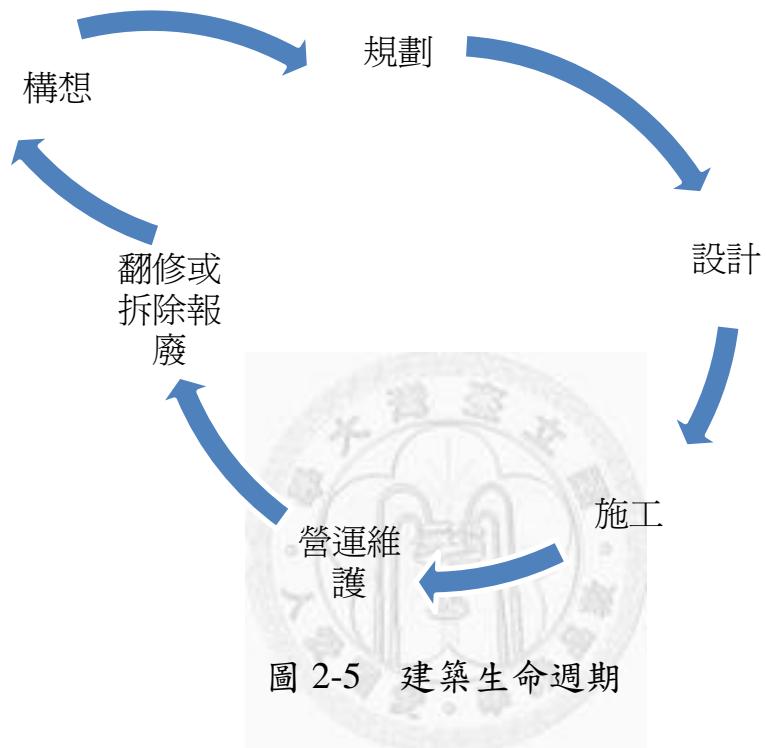


表 2-1 生命週期各項費用分布(蕭樂同，2004)

項目	名稱	費用百分比 (%)
設計企畫費	建設企畫費、現地調查費、用地關係費、敷地整理費、設計費、評價費、不動產取得稅、特別土地保有稅、登錄免許稅、開業權利金	0.4
建設費	公事費、公事管理費、環境對策費、開業準備費、不動產取得稅、登錄免許稅、事業所稅	15.8
維護管理費	建物管理、設備管理、環境衛生管理、清掃管理、保安、警備、修繕更新、管理費用	51.1
營運費	電費、瓦斯費、油料費、水費、污水費	32.3
既定管理費	固定資產稅、都市計畫稅、償卻資產稅、損害保險金、界地費、一般事物費、借入金利息	0.4

以往生命週期中之資訊，階段間資料流通性通常較低，例如當設計階段的資訊到施工階段後，許多資訊會因廠商不同而損失，造成施工與設計介面問題，故讓生命週期間發生或創造之資訊，能在各階段中良好被使用，因此產生建築資訊模型(參見 2.2.2 節)概念。

2.2.2建築資訊模型定義

於一個建築專案中，建築設計資訊在生命週期中為連續，當設計變更時，主要由人工來傳遞訊息更新與圖面之更動。此外各承包商取得 2D 圖檔，依據圖檔進行建物施工時，取得各自獨立之 2D 圖檔，不同團隊往往會因各自背景不同，對圖檔有不同解讀。專案執行過程中資訊傳遞不良與各個團隊溝通困難，會直接影響營建成本的增加，還有營運績效與安全議題等產生。面對營建業使用 2D 圖檔與 3D 檔案衍生之困難，目前已有一種新概念之設計，試圖改進單純使用傳統 2D 工作之不足，此概念稱建築資訊模型(Building Information Model (Process); BIM)，建築資訊模型為土木建築業資訊發展之最新重心，按照 Autodesk 公司的解釋，BIM 為完整之過程，建立可靠且統一之資訊，包含建築設計、建造至營運之資訊。藉由使用 BIM，建築師、工程師、承包商與業主易共同作業、數位化設計與紀錄檔案；BIM 資訊更可提供準確視覺化模擬，分析性能、外觀與成本(Autodesk whitepaper, 2009)；除 Autodesk 公司之定義外，其他根據不同之情境各自給予的定義。建築資訊方法為一種管理建築生命週期中所有資訊之概念，進而產生建築資訊模型(Building Information Mode (Mode); BIM)，此模型包含建築幾何、空間關係、地理訊息、數量性質與建築構件。Renaud Vanlande 等人於 2007 年認為(Vanlande, 2008)，BIM 為一種產生、保存、管理、交換和分享建築資訊，並提供利用和合作溝通。避免傳統上資訊散落，收集整合不易的問題，也提供進一步資訊加值應用的可能性，綜合以上幾種定義，可以歸納出：

- BIM 為一種從建築生命週期各階段收集數位化資訊，包含建築幾何、空間關係、地理資訊、與相關於建築各種資訊之過程。
- BIM 可被利用於各種分析，且增加效率與節省成本。
- BIM 串連建築生命週期中各個介面之資訊，且達到共同作業之效果。
- BIM 之過程必產出一個 BIM 之模型，此模型記錄 BIM 過程所收集之資訊，且提供分析和與使用。

2.2.3 CAD 與 BIM 技術發展情況

BIM 之概念於許多年前已被提出，並且有部分商用電腦軟體支援，但直到近幾年電腦硬體之進步，此類工具才慢慢被普遍使用。由於電腦科技之成熟，許多視覺化工具廣泛使用，並且由 2D 圖面與 3D 模型之視覺化工具逐漸發展至 4D、甚至是多維 nD 之模型運用。BIM 視覺工具技術的發展大致可分為 (康仕仲 等人，2009):

- 1963-1970: 電腦圖學之興起，第一個標示語言 GML 產生、第一台彩色螢幕上市。
- 1981-1990: 商業公司開始提供電腦標準化平台與電腦繪圖標準，微軟開發 MS-DOS(1981)，AutoCAD1.0 在 DOS 模式下開發(1982)，麥金塔系列電腦創立(1984)，微軟開發 Microsoft windows 2000(1985)，AutoCAD 出現圖形化介面(1988)，微軟開發 Windows3.0(1990)，MicroStation 發表第一個 3 維設計的商用繪圖工具(1986)，LightWave New Tek 開發一款專業三維繪圖軟體(1990)。

- 1991-2000:德國公司 Maxon Computer 開發三維繪圖軟體 FastRay(1991) , Auto CAD12 上市(1992) , OpenGL 提供跨程式語言及平台編程介面 , 提出 GDI 指令(1992) , 制定網路描述三維圖形格式 VRML1.0(1994) , W3C 組織正式公布 XML 語發標準(1998) , Blender 上市(1998) , CommonPoint 開發 OpSim(1999) , X3D 標準統一網路三維標準。
- 2000-2004:Intergraph 公司發展 SmartPlant(2000) , Autodesk 發展 Revit(2000) , Civil 3D 上市(2003) , Bentley 開發 MicroStation TriForma V8 。
- 2005-現今:Microsoft XNA 技術發表(2006) , Windows Presentation Foundation 技術發表(2006) , 3Ds MAX 推出新版 3Ds MAX10.0(2007) , OpenGL 更新 OpenGL3.0(2008) , 微軟開發 Windows7(2009) 。

BIM 視覺化工具與相關科技發展快速，原本 2D 工具轉換至 3D 工具耗時較長，然而隨著科技進步，目前以 3D 為視覺化基礎之軟體已經非常成熟。因應越來越複雜之建築結構與機能，以及工程所牽涉之複雜介面管理問題，更多工程資訊與視覺化工具整合，使用建築資訊模型來展示設計概念，並且與其他成員作協同工作也廣泛被使用。

2.3. BIM 技術之協同設計審查應用

由前面小節可知協同之方式與分類，從傳統工程合作開始就存在協同作業。隨近代科技之進步，材料工法愈加精密，建築風格也漸趨多元，專案中資訊較傳統時期更多；而且資訊是散落於各個設計者或者施工單位，讓資訊量以及整合之工作量，早已不是傳統單人控制專案方式能負擔。因此 Fruchter (1999)提出了解決方式，利用許多不同的工具，讓 3D 模型連結網頁說明，讓每個建築物件都可以對應到一個或多個資料頁面，資料頁面也記載物件之相關特性與工程資料，並且程用的應用在協同作業上面，此概念慢慢形成了 BIM，整合在 3D 模型工具中，利用 BIM 方式將資訊整合在一起，產生 BIM 模型，可幫助對專案之掌控、集合散落之資訊與視覺化展示；由於 BIM 模型漸漸流行，亦影響協同作業—特別是協同設計。當所有資訊皆彙整於 BIM 模型時，無形中亦達到協同設計之效果，此種協同設計屬於資料整合之方式，此外一種方式屬於代理人方式，此種方式可讓使用者即時對 BIM 操作，並使與會者同步，讓協同作業更加簡單與方便，此兩種方式皆為 BIM 之發展逐漸突顯出來，並改變建築業原本協同作業型態，本研究以 BIM 模型為基礎，運用 BIM 管理之工具，並具有空間呈現、合併資料與更新資訊能力(Eastman et al., 2008)之條件如表 2-2 所示，針對設計審查為目標，提出相對應之程式架構。

表 2-2 BIM 管理工具條件(Eastman et al., 2008)

BIM 管理工具條件	條件說明
空間呈現(Space object support)	能夠從 BIM 模型中，將空間資訊抓取使用並且呈現。
合併資料(Merging capabilities)	可以呈現不同 BIM 系統所設計的資訊，例如 MEP 和 architectural。
更新資訊(Updating)	如果設施更新，必須更新設施資訊，使資訊保持在符合現況的狀態。





第三章 代理者機制架構設計

3.1. 需求分析

依據美國 NIST2002(National Institute of Standards and Technology)年之報告指出，於建築生命週期中因資料溝通不良與資訊傳達錯誤，當年度全美造成 158 億美元損失，如表 3-1 所示 (Gallaher et al., 2004)，足見資料錯誤與溝通不良會產生龐大損失。Fruchter (1999)指出，利用軟體創造出協同溝通環境，能有效的減少資料溝通不良與資訊傳遞錯誤，但同時也指出營建業於協同作業難處在於，不同軟體間的訊息傳遞困難，由於營建業有著複雜的團隊合作特性，但是不同團隊所需之軟體性質大不相同，各自都有自己所使用之軟體，因此當資料整合時，不同的軟體間溝通成為了困難點。

於營建業資訊呈現分類上根據吳翌禎研究指出，目前土木專案資訊依據呈現方式分成：文字說明、平面圖形、立體模型與立體模型加時間四類 (吳翌禎，2007)如表 3-2 所示。協同審查人員對於這四類資訊，依據實務上需求及所遭遇到之困難，大致可歸類於不易理解 3D 空間資訊，或者不易理解他人之描述。為了解決以上問題，T.M. Froese (2005)於 2005 年提出，藉由使用 3D 模型環境，可有效幫助使用者確定需求與辨認議題；而利用協同設計軟體，使用 3D 模型協同審查辨認議題，並非困難之事，但是市面上 3D 軟體大部分無協同設計的功能，故當無協同設計之軟體，要做協同審查空間溝通時，若非請廠商加入協同功能，就得屈就於語音溝通。軟體廠商尚未增加其協同作業之能力前，是否有其他輔助方法能達成一樣之目的，當不同軟體間要協同作業，常因所屬系統不同而無法達成，相關解決方案之提出及分析於下幾章節進行分析與說明。

表 3-1 生命週期溝通不良損失(單位:百萬)(Gallaher et al., 2004)

	設計階段	建造階段	營運維護	總共
設計師與工程師	1007.2	147.0	15.7	1169.9
承包廠商	485.9	1265.3	50.4	1801.6
業主與管理者	442.4	1762.2	0	2204.6
特別需求廠商	722.8	898.0	9027.2	10648.0
總共	2658.3	4072.4	9093.3	15882.4

表 3-2 營建資訊分類(吳翌禎, 2007)

分類	主要描述方式	系統特性
一維	文字說明	使用文字、數字及表格來描述或是條列計畫資訊，並透過文字格式來傳遞資訊，主要以資料庫及檔案格式來儲存及管理資訊。如傳統之計畫網站及資訊管理系統。
二維	平面圖形	主要利用平面圖形及輔助文字來說明計畫資料、彙整統計結果及作業流程等等之相關資訊。如地理資訊系統、2D CAD 系統、ERP 系統、MS Project 及P3 系統。
三維	立體模型	讓工程相關人員可以透過立體模型的展現，詳細瞭解實際工程介面之情形及衝突狀況，減少錯誤。如3DCAD 系統、虛擬實境系統。
四維	立體模型 + 時程	四維特性系統的概念主要是將規劃設計階段之三維立體模型與施工階段之時程資料加以連結，其特性讓使用者能夠預先地動態模擬施工過程，並從中了解工程潛在性危機及問題，以利及早處理。如4D CAD 系統。

3.2. 代理人設計

本節介紹一種軟體架構，此種架構為本研究之重要構成元素，以下逐一介紹開放代理人模式起源以與本研究應用之狀況。

3.2.1 開放代理人架構(Open Agent Architecture)簡介

代理人(Agent)對軟體而言為經常使用到之機制，特別於協同作業上面，Nwana(1996)指出軟體之代理人分三個面向討論，依據三種因素去討論可將代理人分成四種如圖 3-1 所示，本研究基於使用上以與需求分析，去除學習(Learn)的因素，故最後剩下代理人為本研究所探討之協同代理人，然後再利用開放代理人架構(Open Agent Architecture ;OAA)方式，將代理人組織成有效且容易應用之程式架構。

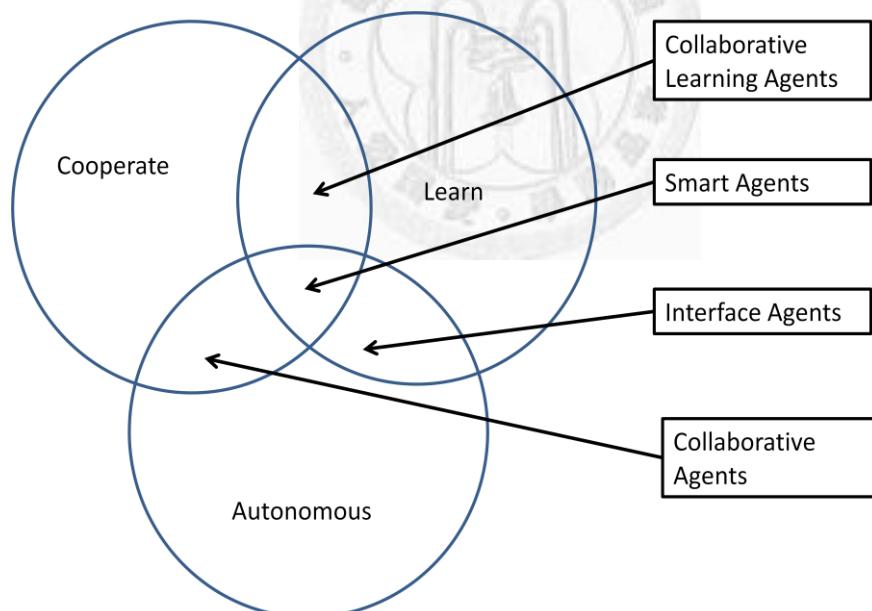


圖 3-1 代理人分類(Nwana, 1996)

Open Agent Architecture(OAA)是由 Stanford Research Institute(SRI)於 1994 年提出(Martin et al., 1999)(Cohen Adam Cheyer et al., 1994)，OAA 將許多不同性質軟體於一個分散環境下（如：網際網路），組織一起協同工作，其建立各個軟體之代理人，並且透過一

個整合代理人，將各個代理人組織成代理人社群；於社群裡各個代理人皆可以獨自運行，另外一方面也可隨著整合代理人所通知事件做出相對應之反應。OAA 架構以協調代理人(Facilitator Agent)為中心，並且採用一種自定之語言 Interagent Communication Language(ICL)，其目的為代理人間溝通，協調通訊各個軟體的工作，是代理人間通訊與相互作用的樞紐，但協調代理人只有接受請求或發布請求，並不集權控制多個代理人工作流程。（The Open Agent Architecture, 2010）案例 OAA 架構圖如圖 3-2 所示。

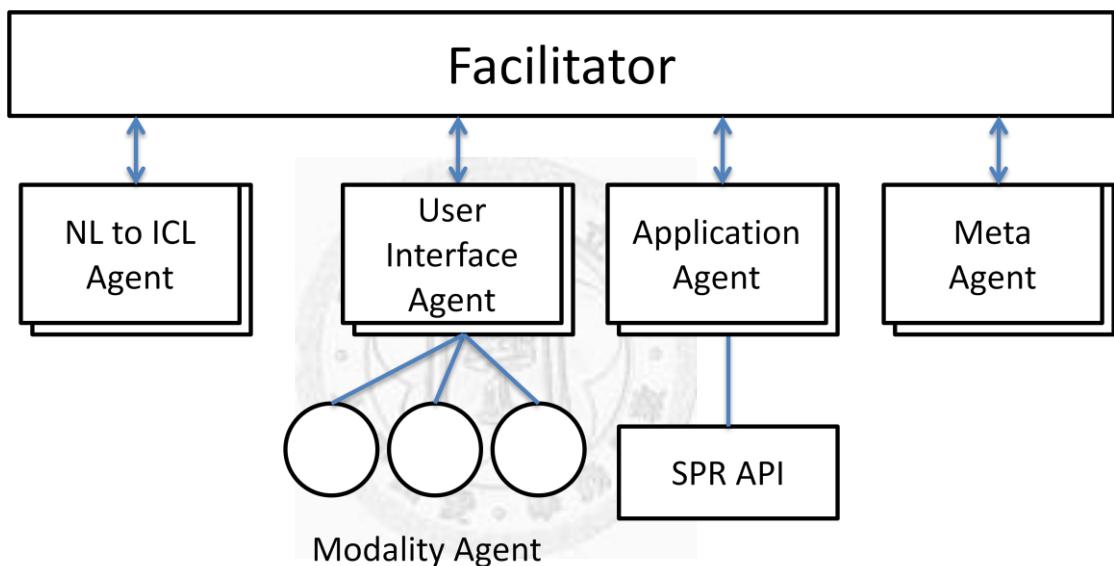


圖 3-2 開放代理人架構(The Open Agent Architecture, 2010)

3.2.2BIM 代理人應用

本研究將應用 OAA 之代理人系統作為基礎，目標將許多 BIM 軟體透過 API，創造各個 Agent，並且利用 BIM Agent 為整合代理人，創造一個 BIM Agent 社群，讓任何用戶端只要呼叫 BIM Agent，可通知相對應的 BIM 軟體 Agent 做出動作。本研究 OAA 架構圖如圖 3-3 所示。

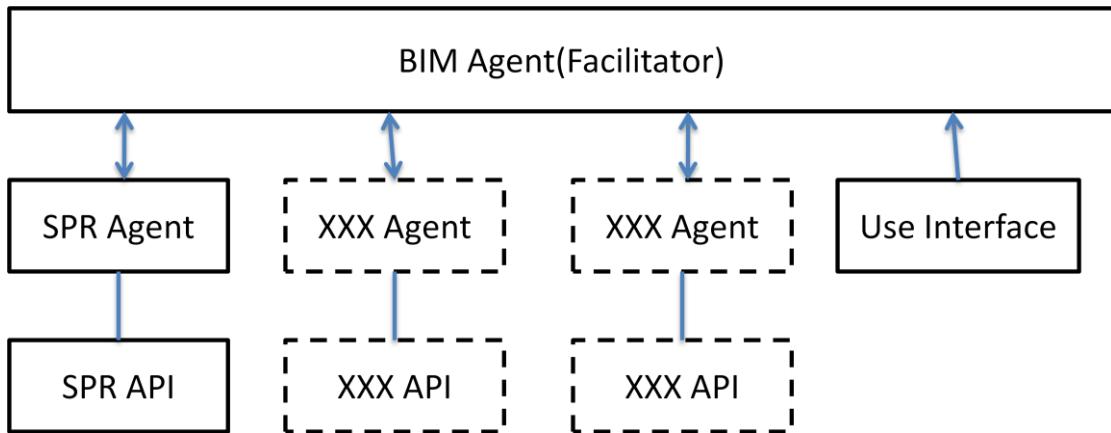


圖 3-3 BIM.RA 開放代理者

BIM Agent 中 Agent 如要能運作，需完成以下幾點事項。首先需建立整合代理人，向整合代理人註冊並且告知整合代理人那些事情可自己解決，此工作於 Agent 被生成時就要完成。本研究利用設計模式中之工廠模式 4.3.2 所提，向代理人(也就是工廠)註冊，利用 BIM 檔案副檔名，註冊其操作的檔案類型，之後於藉由協調代理人發送與調配工作，使以下的代理人完成相對應之命令。



3.3. API 設計

應用程式介面(Application Programming Interface，簡稱 API)，軟體系統不同組成部分銜接約定。因軟體越來越複雜，故將程式分成小的組成，因此各個組成溝通的協定變得很重要。良好界面設計可以降低系統各部分相互依賴，提高組成整個整合性，降低組成組成間相依程度，從而提高系統維護性與擴充套件性。

代理者架構下協調代理者須與不同代理者做溝通，因此在 OAA 架構下有自定一種語言 Interagent Communication Language(ICL)，其目的為代理人間溝通，協調通訊各個軟體的工作，本研究雖然採用 OAA 程式架構概念，但不採用 ICL 語言機制。由於 BIM 軟體就性質而言差異並非很大，故不需要使用超越各種不同類型軟體之語言，可制定一套指令與錯誤訊息回應機制，讓所有代理者皆能接受相同指令，雖然各自處理方式不同，但是接受相同指令會有相同成果。例如當代理者 A 與 B 分別皆收到協調代理者需要關閉程式指令，A 與 B 關閉程式之順序步驟不同，但最後 A 與 B 都關閉了程式。本研究就目前使用之動作與指令區分為三類，基本 API、高階 API 與伺服器溝通 API，基本 API 為較超越應用範圍，像是 BIM 審查基礎的功能，例如開啟圖層、圖層上色與視角設定，並不限定使用範圍，高階 API 為組合數個或一個基礎 API 組合成使用者之功能，例如設定圖層然後著色接著設定視角位置，為一個高階 API 的動作，而伺服器溝通 API 負責處理伺服器與各電腦間關係，以下幾小節會逐一介紹各 API 與應用方式。

3.3.1 基本 API 設計

BIM.RA 之基礎 API 開發工作，目前歸納了 6 個 API 如表 3-3 所示，不管 BIM 軟體為何，只要能夠滿足這 6 個 API 目的，即可使用於本研究程式之下。若欲查詳細資料請參考附錄 A: BIM.RA 基本 API

表 3-3 基本 API 分類

程式介面	功能
LoadBim(String Filename);	開啟 BIM 檔案。
CloseBim();	關閉 BIM 程式。
DisplayDefine(String name, String color);	定義一個圖層並且給定名字與使用的顏色。
DisplayDelete (String name);	刪除圖層。
DisplaySet (String name , String Show);	圖層中設定顯示或者不顯示。
RangeFit();	視角設成剛好可看見所有顯示物件。
ViewUpdate();	更新畫面

3.3.2 高階 API 設計

BIM.RA 之高階 API 開發工作，目前初步實作 11 個介面與伺服器網頁程式，每一個介面皆由基礎 API 組合而來，再配合一個伺服器網頁程式，利用伺服器網頁程式接收資料或指令，驅動本機 SPR 對應之動作，11 個介面如表 3-4 所示，使用者只使用以下幾個介面即可。

若欲查詳細資料請參考附錄 B: BIM.RA 高階 API

表 3-4 高階 API 分類

程式介面	功能
HighLightOneItem(String Filename,String Linkages);	對於一個物件顯示特殊顏色，讓使用者快速了解在哪裡。
HighLightSystemItems(String Filename,String[] Linkages);	對於很多物件顯示特殊顏色，讓使用者快速了解在哪裡。
HighLightRangeFitOneItem(String Filename,String Linkages);	對於一個物件顯示特殊顏色，並且將視角設成剛好可看見所有顯示物件。
HighLightRangeFitSystemItems(String Filename,String[] Linkages);	對於很多物件顯示特殊顏色，並且將視角設成剛好可看見所有顯示物件。
HighLightOnlyOneItem(String Filename,String Linkages);	對於一個物件顯示特殊顏色，其他不相關的物件隱藏起來，讓使用者快速了解在哪裡。
HighLightOnlySystemItems(String Filename,String[] Linkages);	對於很多物件顯示特殊顏色，其他不相關的物件隱藏起來，讓使用者快速了解在哪裡。
HighLightRangeFitOnlyOneItem(String Filename,String Linkages);	對於一個物件顯示特殊顏色，其他不相關的物件隱藏起來，並且將視角設成剛好可看見所有顯示物件。
HighLightRangeFitOnlySystemItems(St ring Filename,String[] Linkages);	對於很多物件顯示特殊顏色，其他不相關的物件隱藏起來，並且將視角設成剛好可看見所有顯示物件。

3.3.3 Server 與 BIM.CS API 設計

除 BIM.RA 中 API 之外，協同伺服器上亦必須具有傳送訊息之 API，以及 BIM.CS 使用於伺服器端專門處理參與人員之 API，並且寫入 XML(參考 4.4.2)檔案中如圖 3-4 所示，然後利用 API 與 XML 檔案配合，才能達到協作審查的要求，以下將列表並解釋其配合 BIM.RA 的 API 如表 3-5 所示。若欲查詳細資料請參考附錄 C:Server 與 BIM.CS API。

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
- <Root>
  - <Interval>
    <name>SP1</name>
    <address>140.112.13.93</address>
    <Online>true</Online>
  </Interval>
  - <Interval>
    <name>SP2</name>
    <address>140.112.12.29</address>
    <Online>true</Online>
  </Interval>
  - <Interval>
    <name>SP3</name>
    <address>140.112.12.30</address>
    <Online>true</Online>
  </Interval>
</Root>
```

圖 3-4 Server XML 檔案

表 3-5 BIM.CS API

程式介面	功能
Server api:	
broadcastAction(String Action, String Adress);	把操作廣播給各與會者，做同步之用。
BIM.CS api:	
Join(String name, String Online, String Adress);	加入協同作業的討論。
Leave(String name)	離開協同作業的討論。
XMLreader();	讀入協同資料的 XML
XMLwriter();	寫入協同資料的 XML
clearAllPerson();	清除所有與會者資料。
GetPerson();	其他的程式獲得協同資料

3.4. 本章小結

本章節依據一開始之需求分析，將需求區分為幫助溝通與不同軟體間協同作業兩種需求。為解決兩種需求，本研究使用 3D 模型之展示方式幫助溝通，創造代理者機制來解決不同軟體間協同作業，而為了協調代理者運作，於代理者間創造了共同 API；API 區分為基本 API 與高階 API，由於目前 API 只針對協同設計審查需求所設計，所以對於其他目的應要增加基礎 API 來滿足；而對於不同軟體間之協同作業需求，本研究於程式架構的彈性以與擴充性方面，皆考慮不同軟體間協同作業，目前完成程式架構實作與驗證，可預期加入其他代理者之實作為可行。

第四章 BIM.RA 與 BIM.CS 的設計與實作

本章假設 BIM 模型資料完整且於統一格式之下，設計 BIM.RA 與 BIM.CS，首先介紹程式概念並與現在技術做比較，然後討論現行兩種用戶端架構(精簡型用戶端與完整型用戶端)優缺點，接著介紹主要設計概念-複合式協同用戶端，最後講解技術細節部分，逐一介紹運用到的技術。本章中提到之應用程式，為利用模型軟體加值開發之相關軟體，例如 Construction Director 時程管理系統(Hsieh, 2006)與 VisPMIS 視覺化專案管理系統(吳翌禎，2007)。

4.1. 設計概念與分析

本小節介紹設計概念，目的是為了突破目前之問題，問題產生於目前營建專案龐大且複雜之資料，資料的統一上常常會有問題，另外 BIM 軟體的使用端架構與 BIM 軟體眾多，不同設計團隊擁有不同 BIM 軟體，因此協同審查上面，變的麻煩需要互相遷就，以下章節把困難之處整理成三點並提出相對應之設計概念，最後將目前可達成目標之軟體分類做比較。

4.1.1 設計難題

本研究之系統開發過程中，整理了三個困難點，分別是設計階段各個部門資料不統一、BIM 軟體眾多與 BIM 軟體不具網路操控能力，以下會逐一介紹，最後介紹相對應之設計概念。

首先是設計階段各個部門資料不統一，營建專案中不同之設計團隊於設計階段，各自取得專案資料，設計成模型發布給不同設計團隊，不同團隊因各自背景不同，對於模型有不同解讀方式。因設計階段常常需要變更設計，資料散落於各團隊，若是沒有把營建專案資訊統一，

會造成專案設計過程中資訊傳遞不良與資訊不正確，會直接影響營建成本之增加，還有設計品質與安全議題產生。

其次是 BIM 軟體眾多，營建專案中由許多不同團隊一起合作，然而各團隊之性質與特性不同，所選用之 BIM 工具各有差別，因此將產生不同格式之模型，目前業界方式利用一個能開啟各類模型之 review 程式(如 SmartPlant review)，於審查中觀看模型進行討論，但是 review 程式雖然可以觀看模型，卻不能改動檔案，若是要改動檔案，必須回到當初設計團所使用之 BIM 軟體上進行，因此如何解決不同團隊不同模型，可利用驅動各自的 BIM 軟體，讀取統一格式 BIM 檔案，達到展示之效果，為設計程式架構時重要之考慮因素。

最後一點是，目前 BIM 軟體不具有網路操控之能力，目前 BIM 軟體都是以完整型用戶端之方式進行操作，各個 BIM 軟體所設計之檔案，於個別電腦中儲存，因此在 BIM 資訊之統一上面臨很大困難，但是若是應用精簡型用戶端之方式，來做整體架構會面臨 3D 模型展示效果差，與伺服器等級須提升之問題，因此在設計上我們考慮，在展示 3D 模型上必須有完整型用戶端之能力，而在必要時候，具有精簡型用戶端之網路操控之能力，因此設計複合式使用端。

以上所整理三個難題，利用程式 BIM.RA 解決其中兩個問題，而 BIM.RA 中運用到了兩個主要之設計概念：代理者機制(參照第三章)與複合式使用端(參照 4.2.3 節)，分別解決了 BIM 軟體眾多與 BIM 軟體不具有網路操控能力之問題。再加上完整之 BIM 模型解決設計階段各個部門資料不統一之問題，完整之 BIM 模型是指利用 BIM 之方法，管理營建專案資料，並產生出 BIM 模型，此 BIM 模型具有資料完整且具有統一格式之特性(參照 2.2.2 節)。

藉由 BIM.RA 與 BIM 模型以解決三大難題，最後再配合 BIM.CS 在伺服器端，管理與紀錄與會者資料，來達成協同設計審查之目的。

4.1.2 比較分析

為了達成協同審查之效果，目前市面上有許多種方式，其中分成兩大類，第一類為遠端桌面方式第二類為虛擬空間。

遠端桌面系統是一種讓使用者操作位於遠處電腦的方式，這種系統藉由傳輸桌面影像，依據使用者的操作傳輸部分更新畫面，相對來說資料傳輸量就少得多軟體如，TeamViewer、Skype 畫面分享、Gogrokm 與 NetMeeting。

虛擬空間系統是一種空間概念，把資訊放入空間中，讓進入此空間之人，可以看到相同之資料，如 BIMserver、GameBase 軟體等等。

第一類之優點在於技術成熟已廣泛被運用，但是在控制之過程中，只有一個使用者可以操控電腦，對於營建業來說，各個部門所關心議題皆不相同，若是只有一個人能控制，並不能滿足各個部門之需求，而本研究之架構特色為：先同步各個與會者畫面，讓各個與會者對於討論議題已有所掌握，之後給與各個與會者隨意操控 BIM 軟體之自由度，不用畫面都被人所控制，為了要做到這目的，設計了複合式使用端，在設計上每個人都為 Fat client，可獨立操作 BIM 軟體，當要同步時候有能利用 Thin client 被同步，因此本研究可以配合營建專案特性使用。

第二類之優點在於資料都統一在虛擬空間中，與會人能夠於虛擬空間中隨意觀看，因此有各自之自由度，能符合營建專案之特性，但是由於目前營建專案所使用之檔案模型種類相當多，不同部門利用不同工具所產生之模型，還無法有共同之空間，因此要架設共同之虛擬

空間困難性較高，因此本研究之架構提出整合各種工具之代理人技術，並且已變動最小之方式，達到協同審查之目的，比起建設共同虛擬空間較不複雜。

4.2. BIM 協同操作之種類

個人電腦能力大大提升，以前需要較好伺服器等級配備，才可以達成 3D 模擬，現已逐步邁向個人電腦皆可模擬，許多應用與方法應此產生，網路 3D 模擬亦為其中熱門一種。目前市場上網路 3D 模擬技術分兩種，一種為加強瀏覽器之能力，使瀏覽器成為具有瀏覽 3D 之能力，但此種方法在技術門檻上較高，目前以 Google 公司和微軟公司為主要發展者；另一種為藉由伺服器端之處理而後傳送圖檔至使用者端，優點為不造成使用者端電腦太多負擔，而缺點為處理之項目增加，對伺服器之設備要求提高，而成本會大幅提高。此兩方式可歸類為大型主機與單機作業兩種模式，以下會就此兩種作介紹，最後將介紹結合兩種之優點而產生複合式用戶端。

4.2.1 精簡型用戶端(Thin client)架構定義

精簡型用戶端 (Thin client ;精簡型電腦)，網路應用程式之特色為強調伺服器端之處理，應用程式不需在工作站上作執行或儲存之動作。

精簡型用戶端操作之原理分兩種，第一種為終端機服務(terminal service，又稱 Dos application):特色為將資料與計算能力，集中於後端之大型主機，前端電腦等級極低，因前端電腦只需要執行連線程式，於伺服器端建立一個遠端工作區，用戶端連線程式，將輸入(按滑鼠、敲鍵盤)傳到伺服器端，而後顯示出伺服器端應用程式介面之影像。

此種模式就像把用戶端當作一台顯示終端機，把輸入和輸出資訊由伺服器端進出。

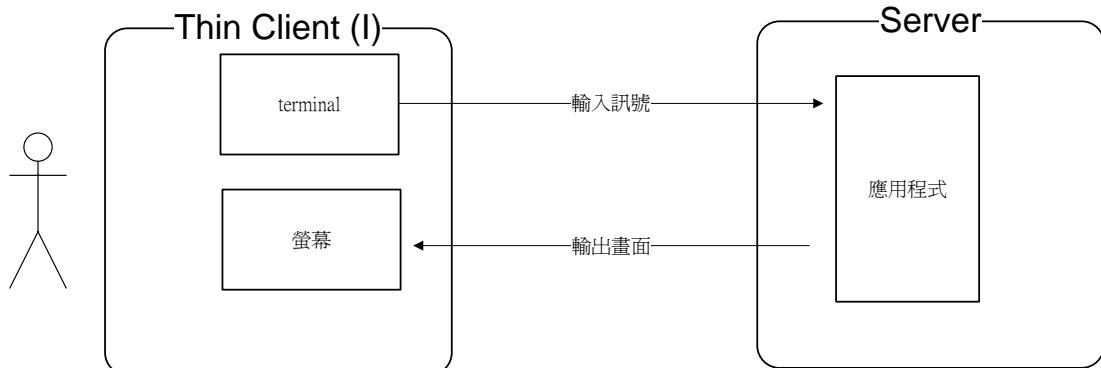


圖 4-1 Thin Client(I)

第二種就是利用小型瀏覽器(Brower applet，又稱 Web App)，此種軟體無論在什麼平台或電腦上，只要有一個簡單瀏覽器，即可執行放在伺服器端各種應用程式功能，而運作上幾乎完全都是透過網路來進行，用戶端在執行功能時是使用非同步之方式來呼叫伺服器上之功能。

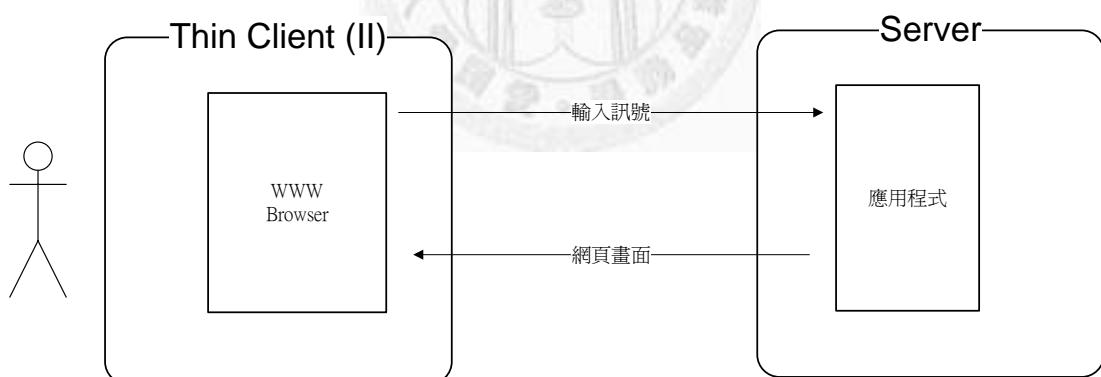


圖 4-2 Thin Client(II)

精簡型用戶端之優點：

1. 容易安裝：因應用程式之功能完全存在伺服器端，用戶端完全不需要進行安裝軟體之工作。
2. 容易管理與更新：若有需要更新應用程式，將只需要更新伺服器之功能不需改變用戶端之程式。

3. 攜帶維修方便:因用戶端所需電腦等級極低，故可用很輕便和簡單電腦完成工作。
4. 協同工作:因資料都於伺服器端，資料共享可協同作業。

精簡型用戶端之缺點:

1. 須有網路:因此種用戶端高度依賴網路與伺服器溝通，若無網路則失去工作能力。
2. 使用者操作較差:與一般 PC 比較，此種用戶端對於使用者介面限制很多，瀏覽器除能瀏覽基本文件與圖片，若需要更多變化或者多媒體聲光效果，瀏覽器本身無法提供足夠支援。
3. 伺服器負擔過大:當過多用戶端在使用時，常會影響使用效益，故伺服器條件亦十分重要。

4.2.2 完整型用戶端(Fat client)架構定義

完整型用戶端通常稱為個人電腦 (Person Computer ; PC)，其特色為所有程式與資料都會裝至同一台電腦之中，故程式運作時會使用同步方式與底層資料進行溝通，處理大部分圖形使用者介面之顯示與控制邏輯，可單獨作業不需依靠其他電腦，故需要較高階與完整之設備如硬碟、光碟機等。隨著網路發達，常會有資料存放於網路上，在此則是使用提取資料方式，將資料下載回來之後進行處理，此方式又稱為兩層主從式架構(Fat client)，兩層式架構中，資料本身於伺服器中不會經由任何處理，使所有資料處理動作存於用戶端，用戶端能於處理資料方式上具彈性。

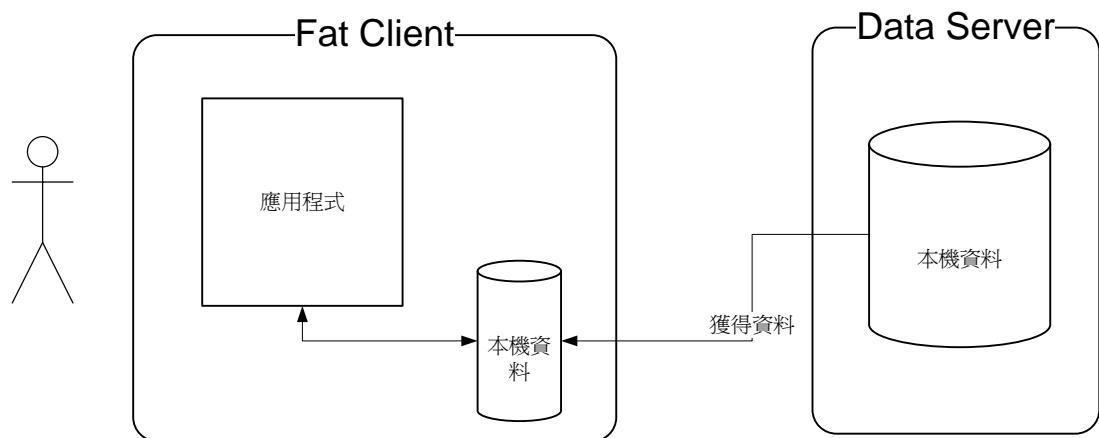


圖 4-3 Fat Client

完整型用戶端之優點:

1. 用戶端使用者介面極為豐富:因應用程式都在一台電腦上，故能做到較好之使用者介面。
2. 資料處理效率高:資料與應用程式皆在一起，故能夠即時回應使用者對資料操作之結果。

完整型用戶端之缺點:

1. 協同作業不易:由於各自有獨自資料，無法即時統一與分享資料。
2. 安裝程式易失敗:因每台電腦皆可自行運作，但環境皆不同，故當安裝新程式時，容易因環境不同而失敗。

4.2.3複合式用戶端架構定義

過去大型主機第一型終端機介面傳統架構，至完整型用戶端的 PC 普遍應用，接著逐漸回到大型主機第二型瀏覽器的架構，但兩種型態的架構，優缺點能夠互補，大型主機優點為個人 PC 缺點，而 PC 之優點亦為大型主機之缺點，故最近擷取個別之優點並改善舊技術之缺點，產生複合式用戶端架構(簡稱複合端)，複合端同時具網路操控與應用程式操控之平台。傳統上，Fat client 需完整作業環境，開發條件

亦較嚴苛，若環境完整後，開發速度將較快速，且不會有整合上面之困難；而 Thin client 亦有開發方式複雜、需建立網路連結與所能創造使用者操作經驗較貧乏與 3D 模型展示效果較差之缺憾。複合端架構圖如圖 4-4 所示，於圖中複合端分成兩種使用方法，

(一)利用本機程式如時程排序程式直接使用。

(二)利用資料庫方式，藉由網路來操控。

(一)操作流程為(1)應用程式下達指令->(4)協調代理者依據指令產生動作

(二)操作流程為(2)藉由瀏覽器傳指令給伺服器->(3)伺服器做出相對應動作(展示資料在瀏覽器、傳送指令給協調代理者)->(4)協調代理者依據指令產生動作。

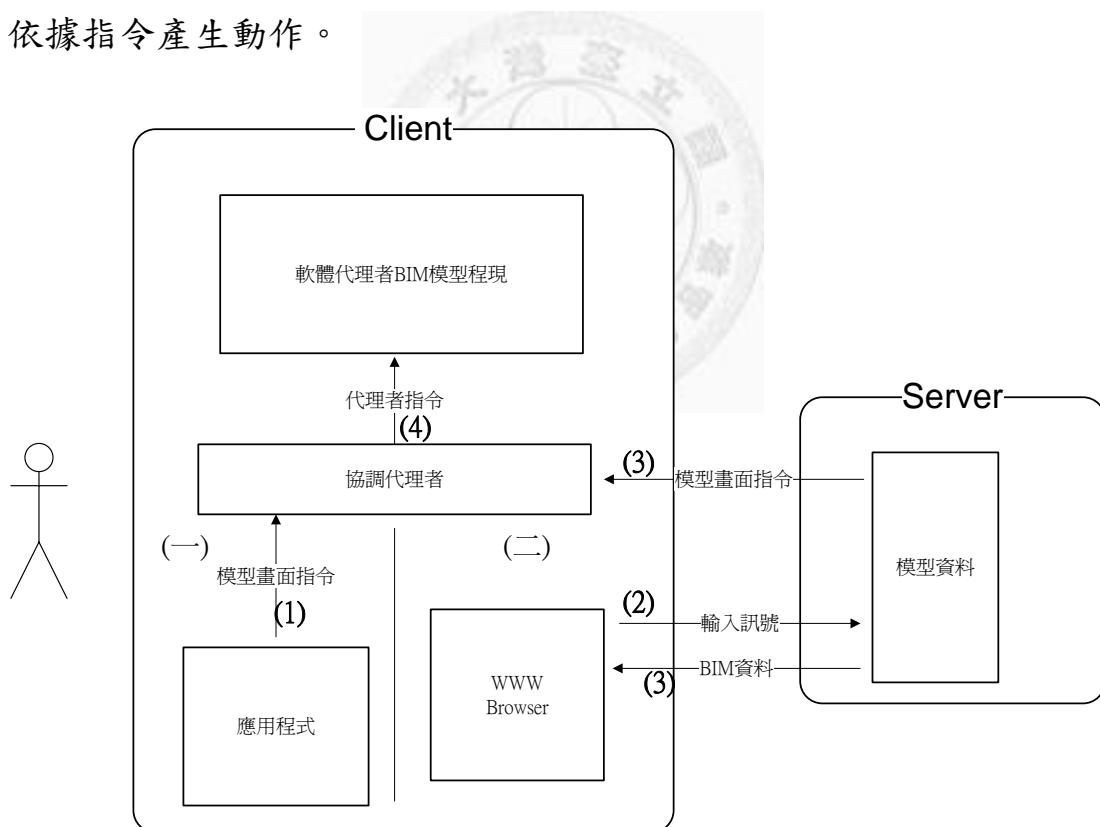


圖 4-4 複合用戶端

複合端之優點:

1. 用 戶 端 使 用 者 介 面 極 為 豐 富 : 因 應 用 程 式 於 本 機 端 , 對 模 型 操 控 能 力 與 展 示 效 果 皆 可 利 用 強 大 PC 資 源 , 不 會 受 限 於 網 路 傳 輸 之 限 制 。
2. 協 同 作 業 容 易 : 因 資 料 於 網 路 上 統 一 存 放 與 分 享 , 不 會 有 版 本 不 一 致 之 問 題 。
3. 容 易 安 裝 : 因 模 型 展 現 功 能 完 全 由 伺 服 器 端 決 定 , 用 戶 端 不 需 執 行 多 餘 之 軟 體 , 即 可 達 成 模 型 展 示 之 功 能 。
4. 伺 服 器 負 擔 小 : 當 許 多 複 合 端 使 用 時 , 不 需 對 模 型 進 行 處 理 , 只 需 對 於 BIM 之 文 字 資 料 作 管 理 , 大 大 減 少 伺 服 器 所 需 之 規 格 等 級 。

複合端缺點:複合式用 戶 端 為 完 整 型 用 戶 端 , 再 加 上 精 簡 型 用 戶 端 的 能 力 , 因 此 會 有 某 些 完 整 型 用 戶 端 之 缺 點 , 例 如 攜 帶 不 方 便 因 BIM 用 戶 端 所 需 電 腦 等 級 尚 須 較 好 等 級 , 故 無 法 用 輕 便 與 簡 單 電 腦 完 成 工 作 , 但 對 BIM 設 計 者 而 言 , 設 計 工 作 會 在 固 定 的 地 方 完 成 居 多 , 故 此 說 這 個 缺 點 對 於 此 研 究 而 言 並 不 是 重 點 。

4.3. 物件導向設計

4.3.1 物件導向技術概念

物件導向技術為一種程式設計思維，主要概念為在程式的每一個物件，皆都具有獨立運作處理資料之能力，並且把處理完成之資料傳給其他物件，故皆可被看成一個小型機器，或各個負有責任之角色，其中三大重點：封裝、繼承與多元性以下逐一介紹

- 封裝(Encapsulation): 將資料與處理資料程序合併包裝於一個單元，區隔出單元與單元間可被使用之特性，及內部特性，把關聯性降到最低，如此一來將不會產生牽一髮而動全身之狀況，而降低互相依賴程度亦等於降低了複雜度，進而方便開發與維護。一般而言模組化與資訊隱藏亦可包含在廣義之封裝定義內。
- 繼承(Inheritance): 由既有之類別中，或者稱基底類別 (base class)，擴充建立一個新類別稱為衍生類別(derived class)。衍生類別具基底類別之特性，且可擴充一些特性，而衍生類別之改變不會影響基底類別，此特性有效提升程式碼的再利用性，設計人員可快速開發，不僅可修改基礎類別，亦可透過繼承與多型，產生不同子類別而適應不同環境與不同需求。
- 多元性(Polymorphism): 亦稱多型是物件導向技術當中重要之技術，簡而言之多元性之目的為讓使用者，依據需求與狀況得到自動獲得最適當之型態，亦同一物件有兩種以上之解釋，當程式設計師呼叫具有繼承關係之類別時，系統會依照類別之種類選取對應的處理方式。多元性大多使用介面(interface)或是使用抽象(Abstract)的類別來實作。

4.3.2 物件導向設計模式

由於 BIM.RA 之功能為經由軟體公司所提供之 API 加以實做，BIM.RA 功能與其自由度因此而受限制，每家廠商皆有所不同，由於必須考量 API 有無提供的相關程式開發介面，並加以規劃共同 API，於是導入物件導向設計模式(Design Patterns)，使 BIM 展式軟體加入或移除 BIM.RA 更加方便與容易，BIM.RA 中運用兩個模式，工廠模式(Factory Method)與單一模式(Singleton)以下將逐一介紹(Gamma et al., 1995)(Gamma et al., 葉秉哲譯, 2001)。

- 工廠模式(Factory Method):於架構上有許多圖台程式，會因為協調代理者之調配與通知而工作，於是當使用者給予協調代理者一個工作後，協調代理者將利用工廠模式自動判斷，並生成一個特定代理者，並且傳指令給代理者(圖台代理者)，然後呼叫圖台完成工作。其優點為易於將 Agent 之物件抽換，依據使用者目的而產生特定代理者，使用這個模式，能夠確定 Agent 有固定的 API，方便開發者對不同圖台之開發。
- 單一模式(Singleton):於架構上有許多圖台程式，但是只有一個協調代理者，確保代理者只有一個物件實體存在，並提供單一存取窗口，讓使用者只對他發送指令，達到良好之管控。

綜合兩個設計模式，BIM.RA 的架構如圖 4-5 所示。

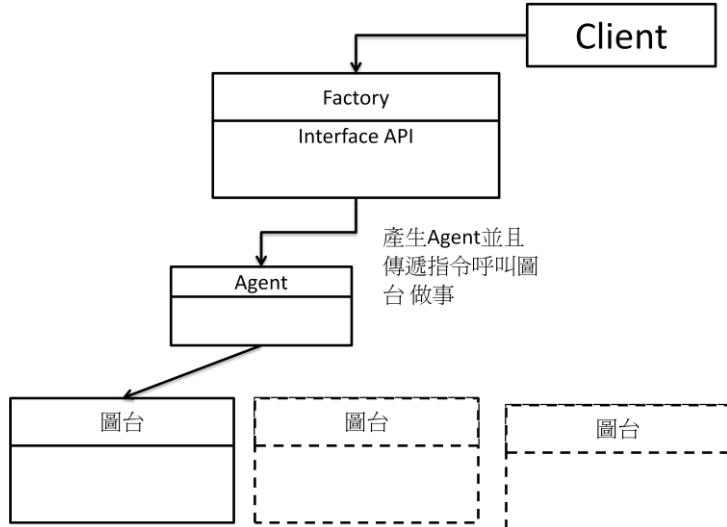


圖 4-5 設計模式架構

4.3.3擴充性

BIM.RA 利用物件導向與設計模組概念，目的是為讓程式本體容易適應不同 3D 軟體，進行擴充與配合使用者。本研究利用 JAVA 與 Smart Plant Review 進行實作，但 BIM.RA 程式架構本身並不限定用任何語言實作，只要語言本身具有物件導向，並且能與網路資源做連結即可，例如 C#、VB 等。於 3D 軟體方面，本身的 API 能夠組合出 3.3 介紹之相同功能 API，即可新增代理者於協調代理人之下，並且交由協調代理人管控。

4.3.4系統使用者

本研究的程式架構應用面廣泛，凡是需要空間概念之溝通，皆可使用，本研究以設計審查為例。於設計審查方面，使用人員為設計審查人員和與會人員，使用人員藉著使用 BIM.RA 進行統合工作，且藉由 BIM.RA 幫助準確且快速瞭解現地資訊，做出適當的人力與機具調配，並討論與解決空間衝突，減少不同設計部門間溝通時間。

4.4. 軟硬體架構

協作審查平台程式之軟體架構，分為程式語言、繪圖套件、與程式架構來說明。

4.4.1 程式語言

本研究於網路介面上，將選擇 Java 為主要語言，原因為 Java 對於網路支持性較高，並且於主流之網路技術上占有一席之地。Java 為一種可以撰寫跨平台應用之物件導向程式語言，由昇陽(Sun Microsystems)公司於 1990 年代初期以 C++ 為程式發展基礎開發，命名為 Oak，卻因有人使用而取名為 JAVA，本來目標於家庭電器等消費性小型系統使用，一開始消費性電子產品並未如預期的發展，團隊幾乎面臨解散危機，之後網路興起帶動 JAVA 轉型，而最重為 JAVA 能在網頁上加上動態文字與動畫，成為當時最炙手可熱之語言。

現今 JAVA 應用區域漸漸廣大，如企業大型網頁應用程式與原本消費性電子產品，如手機、PDA 等皆可由 JAVA 所撰寫。

JAVA 具有下述之優點，對於程式語言而言，為高階語言並且易撰寫，使用一般編輯文字即可開發。由於為高階語言，許多有用函式庫均已開發完成，且軟體為使用上較無版權的問題，JAVA 程式可以執行在不同平台上，不需針對不同之作業平台而對程式做更動，即可執行 JAVA 物件導向之特性，具有較好擴充性與程式碼再利用。

4.4.2 可延伸標示語言(eXtensible Markup Language)

可延伸標記語言(eXtensible Markup Language ; XML)，為一種電腦所能理解之資訊符號。透過此種符號，電腦間可以處理包含各種資訊之文章(World Wide Web Consortium, 2010)。XML 是由 1995 年起於全球資訊網協會 (World Wide Web Consortium ; W3C) 上被提案，於 1998 年 W3C 公布標準，之後在網路上大量被使用，現在已成為網路技術重要的一環，XML 優點有：

- XML 為通用之標籤語言。
- XML 文件具有自我描述之能力。
- XML 文件之有效性為可被認可。

基於以上優點，本研究於記錄參加會議者的資訊時，亦採用 XML 方式記錄，當與會者向伺服器要求加入會議時，伺服器自動會抓取使用者之 IP 與電腦名稱，且加入伺服器中人員資料之 XML 檔案，人員資料定義 3 個標籤如圖 4-6 所示，使用者名稱(name)，使用者所在的電腦 IP(address)，是否在線上(online)，藉由 XML 檔案可了解目前參與人的狀態與 IP 位址。

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
- <Root>
  - <Interval>
    <name>SP1</name>
    <address>140.112.13.■</address>
    <Online>true</Online>
  </Interval>
  - <Interval>
    <name>SP2</name>
    <address>140.112.12.■</address>
    <Online>true</Online>
  </Interval>
  - <Interval>
    <name>SP3</name>
    <address>140.112.12.■</address>
    <Online>true</Online>
  </Interval>
</Root>
```

圖 4-6 XML 檔案

4.4.3繪圖套件 SmartPlant Reviews(SPR)介紹

繪圖套件上，選擇 Intergraph 的 Smart Plant Review(簡稱 SPR)，SPR 為一個強大 3D 展示軟體，於業界上亦被廣泛使用。

SPR 為 Intergraph 公司中 SmartPlant 產品系列中一種，SPR 特性為可包含各種不同 3D 作業之檔案，可同時呈現管線、鋼構、混泥土建築等模型。對設施維護管理階段而言，建築物已經完成，所以不能只單看某些類別的圖面，而以根據想看與否，決定是否顯示，並且有強大之模擬與視覺效果展示建築成果，展現工程狀況，並應用於現地管理，且擁有 API 介面，可開發自己軟體，配合 3D 顯示達到管理效果 (Intergraph, 2010)，例如 Construction director 為一個 4D 管理軟體，是利用 SPR 之 API 達到時程管理 (Hsieh et al., 2006)。

4.4.4程式架構

於程式架構方面，除應用 Java 語言撰寫物件導向之圖台介面，採用 SPR 之 API 部分做為繪圖輸出，還使用了幾個 Java 程式庫做為開發的輔助工具，分別為 Jsp 網路工具與 JNI 原生介面工具，以下將一一說明。

- JavaServer Pages(Jsp):由昇陽公司與許多公司共同建立，JSP 特色為可回應客戶端請求，動態請求生成各種網頁格式，做出互動，以 Java 語言作為指令語言，JSP 網頁為整個伺服器端的 Java 函式庫單元提供了一個介面來服務於網路的應用程式。Jsp 使 Java 代碼與預定動作嵌入靜態頁面當中。
- Java 原生介面(Java Native Interface ; JNI)，為 Java 所提供之一個方法與其他語言相互溝通，目前支援 C、C++ 與組合語言，

通常被用於讓 Java 使用到其他程式所提供之程式庫或作業系統底層功能（呂英瑞，2004）。本研究所使用 JNI 為跨過 Java 虛擬機器，直接與 SPR 程式庫做溝通如圖 4-7 所示。

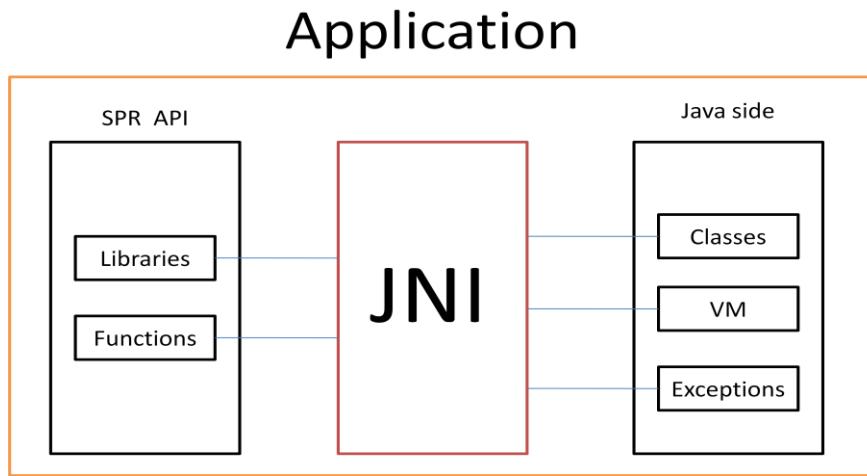


圖 4-7 JNI 程式架構

4.4.5 BIM.RA 與 BIM.CS 使用與運作流程

BIM.CS 與 BIM.RA 並不是視窗類型之程式，BIM.CS 屬函式庫，是讓伺服器端使用，具有操作 XML 資料之能力，包含讀取資料、刪除資料或寫入資料。BIM.CS 使用介紹分成兩大部分，首先為 XML 操作，當選擇加入或刪除某人之指令如圖 4-8 (A) 所示，BIM.CS 會利用 XMLreader(); 讀入在伺服器端資料，作對應操作，完成後再寫回 XML 檔案中。另外一部分為伺服器廣播指令時之流程如圖 4-8(B) 所示，首先選擇指令，而後 BIM.CS 讀取與會者資料，伺服器依據資料內容傳送指令，達到廣播效果。

BIM.RA 為介於在網路資料庫與本地套裝軟體，專門接收資料庫命令並且利用代理者機制，進行動作達成目的。而整個流程一般而言分為幾個步驟如圖 4-9 所示，首先伺服器端會傳送指令，本地之 BIM.RA 接收指令，會先判別是否成功，若未成功將回傳錯誤訊息，

訊息成功接收之後，協調代理人會判斷要給哪代理人任務，若無相對應之代理人，會回傳錯誤訊息給伺服器，而成功配對代理人後，傳給代理人指令，由代理人完成動作。

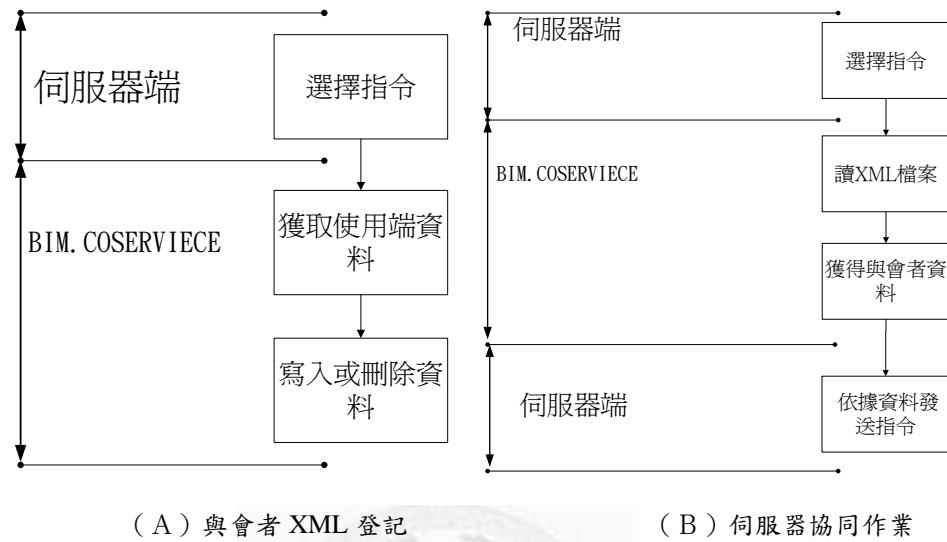


圖 4-8 伺服器端作業流程

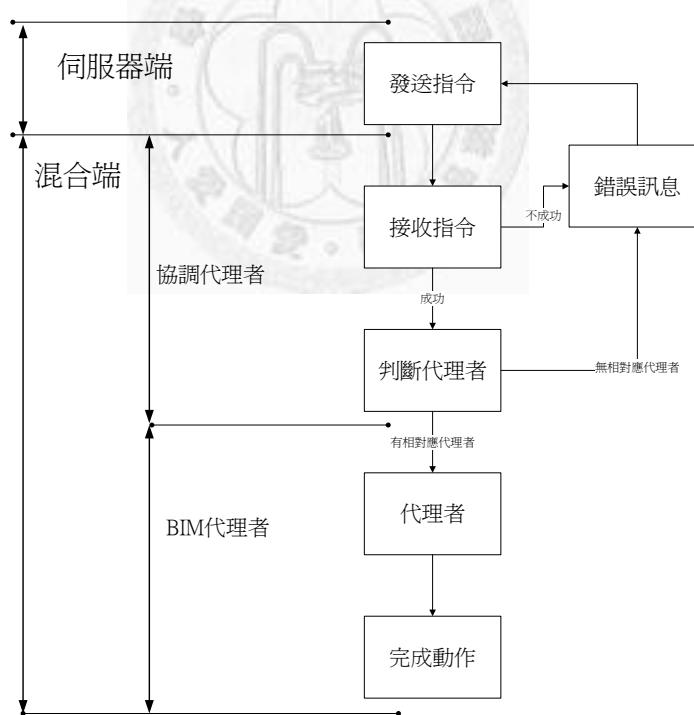


圖 4-9 混合端作業流程



第五章 電腦輔助設計審查展示案例

介紹完本系統架構與實作之兩個程式(BIM.RA 和 BIM.CS)以及相關特性後，本章節以實際模擬多人連線合作審查，展示使用本系統實際連線互動與審查之流程。

本章節分為設計審查案例說明、模型資料、執行流程與結果展示四個部分逐步做敘述，設計審查案例說明，先講解測試案例中各個電腦的關係；模型資料講述所採用之模型背景與特點；執行流程說明運作時之狀況；結果展示表現出此示範案例之結果。

5.1. 設計審查案例說明

本範例由三台位於不同位置之個人電腦與一台資料伺服器，個人電腦分別裝設 BIM.RA 與 BIM.CS 兩個程式，伺服器資料庫裝置 BIM 資料，三台電腦來觀看同一模型，針對一個物件做展示，並協助團隊對有問題物件審查。三台電腦以及伺服器的 IP、Port 與用者名稱如圖 5-1 所示，利用實體 IP 來模擬真實之網際網路狀態。

另外此三台電腦分別代表設備設計師、鋼結構設計師與管線設計師三種系統，如表 5-1 所示，並且在下一小節中會提及三種不同人員需要協調的狀況。

表 5-1 案例說明表

人員	電腦
設備設計師	Computer A
鋼結構設計師	Computer B
管線設計師	Computer C

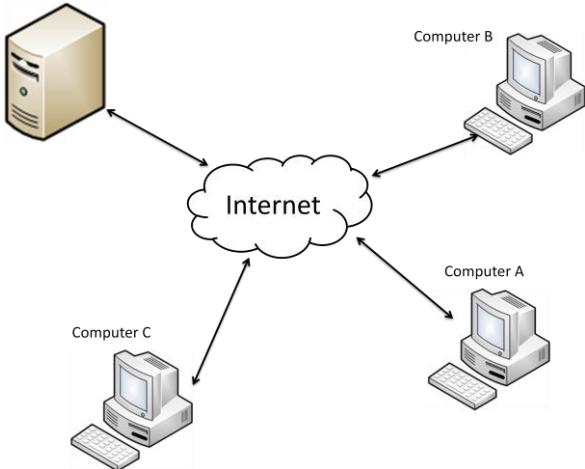


圖 5-1 案例情境

5.2. 模型資料

本研究驗證模型採用實際三分之一石化廠模型如圖 5-2 所示，此模型包含了結構部門、管線部門與設備部門三大部門，其中模型元件數量為 23795，其特性有同樣大小的空間，元件密度遠大於住宅建築、專案規模較龐大與較一般建築工程部門配合更為密切且頻繁。

石化工業因其內容複雜，設計品質要求嚴格，採用 3D 模型進行跨部門整合設計，設計品質因而大幅提高，3D 模型已成為設計工程師不可缺少之工具。然而不同設計部門，分別完成設計模型後，集合於審查時，會發現衝突點，當衝突發生後，目前業界解決的方法是將系統位階化。如表 5-3 所示(資料來源為訪談資深工程師所得)，當衝突發生時，會以位階較大為主，位階相對較小作禮讓，例如：有個衝突為位階 1 之設備與位階 3 之結構衝突，會希望結構讓步，並且重新設計不影響結構安全，而在此過程當中大部份衝突可依此方法解決，但是有些衝突需要設計人員開會討論，才能夠解決。本研究衝突之案例，採用衝突之重大議題，也就是說模型可能有非常多的衝突點，但是有些重大衝突點，不能依據位階來處理之衝突，必須經由開會討論才能進行衝突的排解，以下列了四個重大衝突點如圖 5-3(A) (B) (C) (D) 所示，並詳述於下表 5-2 中。

表 5-2 衝突點說明表

衝突點編號	狀況說明
案例中(A)衝突點	圖裡面 A 為設備之監測器，作為監測是否變形，B 為結構之 H 型鋼，為建築物主之撐設施之用。
案例中(B)衝突點	圖裡面 A 為大管線，於石化場中主為連接設備，B 為結構之 H 型鋼，為建築物主要支撐設備之用。
案例中(C)衝突點	圖裡面 A 鋼結構斜撐，為建築物主要支撐設備之用，B 為機電之電纜槽，主要用於集中機電管線之用。
案例中(D)衝突點	圖裡面 A 為大管線，於石化場中主為連接設備，B 為鋼結構斜撐為建築物輔助支撐設備之用

表 5-3 衝突解決順序表

位階	建築系統	原因
1.	設備	屬於重要設備，變更位置會導致大幅度變更設計規劃。
2.	大管線(>10 英吋)	連接重要設備，例如主要輸油管線，不可輕易改變位置。
3.	結構	關係整個建築物的安全，提供人員操作空間，支撐設備、管線、機電等設施。
4.	管線(<10 英吋)	改變較容易。
5.	機電	改變較容易。

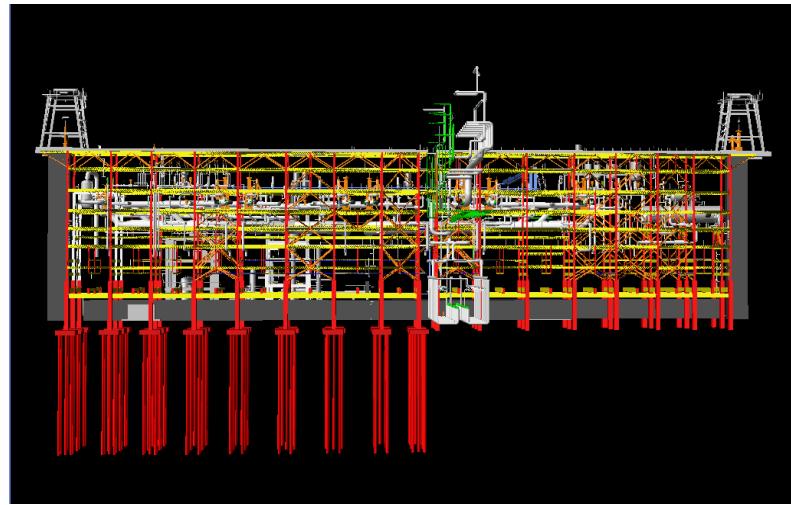
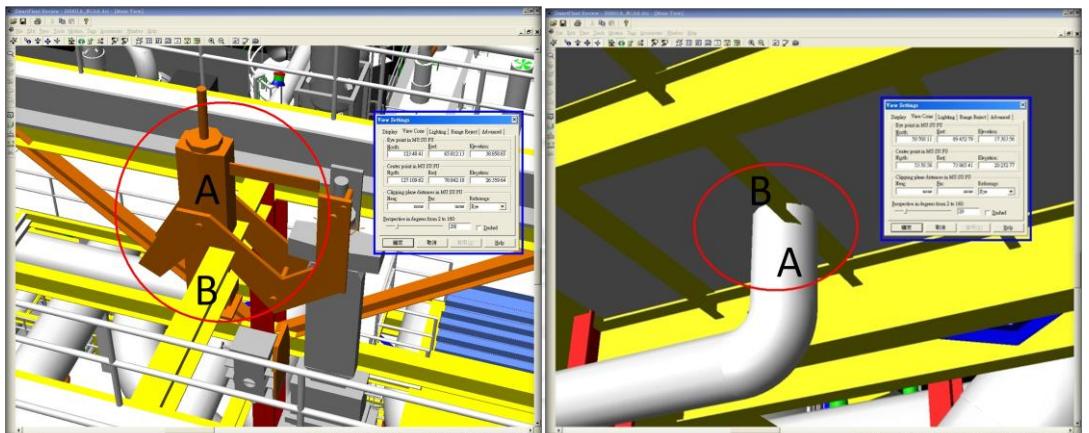
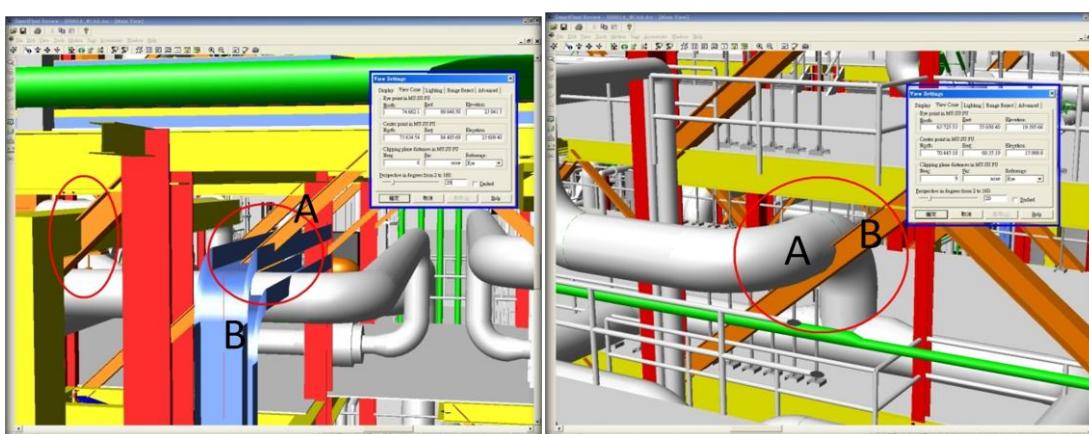


圖 5-2 案例模型



(A)衝突點一

(B)衝突點二



(C)衝突點三

(D)衝突點四

圖 5-3 案例衝突點

5.3. 實行流程與結果展示

5.3.1 操作流程概述

本研究之範例操作流程分成兩部分討論，第一部分為建立三台電腦之關係，第二部分為開始討論之流程。

首先要建立三台電腦關係：

- 開啟網頁畫面如圖 5-4(A)，其中按鈕 Infor 與 Join，其中 Infor 式展示目前電腦跟哪些電腦有連線，Join 加入其他人協同團隊，另外 Delete 可刪除協同團隊中某人。電腦 A、B 與 C 於設定加入協同之畫面設定好(分別為 sp1、sp3 與 sp2)後如圖 5-4(B)，設定主機後回到 Infor 狀態已經完成設定。

Name	IP	主機

Delete

Infor Join

Type	Content
Name	sp1
Host IP	140.112.13.93
Validation	Regenerate

Infor Join

(A) 展現與會資料畫面

(B) 註冊資料畫面

圖 5-4 與會者 XML 登入畫面

- 完成設定後伺服器端會紀錄一個 XML 檔案，如圖 4-6 所示其中逐一介紹 XML 標籤，使用者名稱(name)，使用者所在的電腦 IP(address)，是否在線上(Online)。

接著為討論流程：

案例中討論模型範例中(A)衝突如圖 5-3(A)所示，其中 A 為設備，B 為 H 型鋼，整個流程如圖 5-5 所示，當 Computer A 欲想與其他電腦討論時，會發送指令給伺服器(圖中虛線)，透過伺服器傳送指令給每一台電腦(圖中實線)，達成每一台電腦畫面同步之效果。

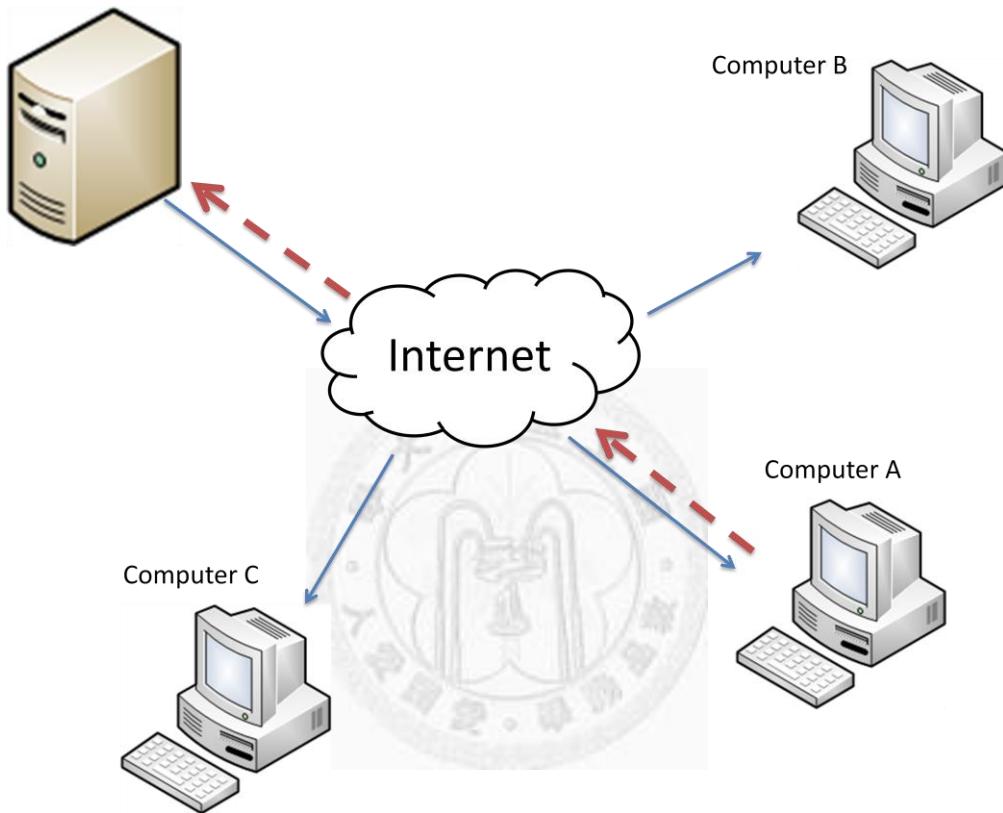


圖 5-5 協同流程

5.3.2 結果展示設定

本研究為突顯衝突問題，採用藍與黃之對比色呈現，利用兩種對比色突顯模型之衝突點，原因是本身模型不同模型檔案，軟體自動會用顏色分別，但由於畫面上呈現的物件太多，顏色雜亂再加上與衝突點相同顏色的物件，也會影響與會者注意力，於是把模型中衝突點之外所有物件，歸類同種顏色而衝突點利用對比色呈現，能幫助與會人員快速了解衝突點的狀況。

首先將整個模型以藍色方式呈現如圖 5-6，然後將衝突之元件以黃色呈現如表 5-4(A)(B)(C)(D)所示，以(A)為例，當 Computer A 發送指令希望看到(A)之突顯衝突點狀態時，伺服器接收到指令後，發送給在討論內所有電腦相同指令，所有電腦之協調代理者接收到指令後，依據代理者機制，顯示如表 5-4(A)之突顯衝突點狀態所示。

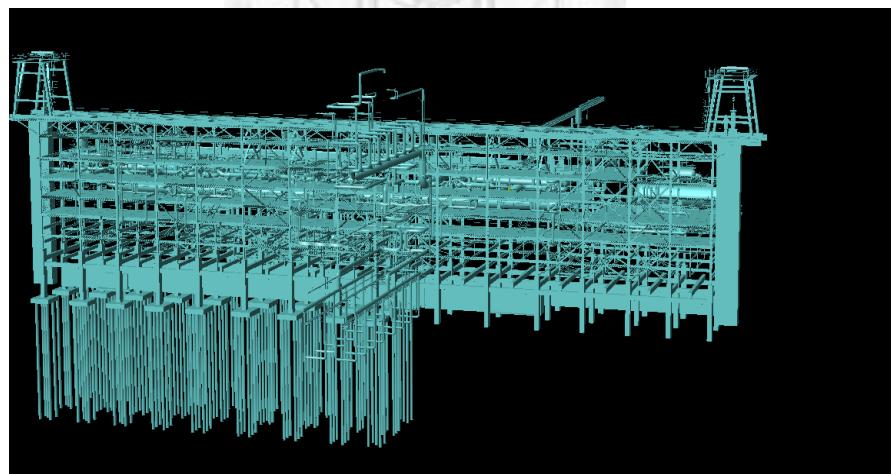
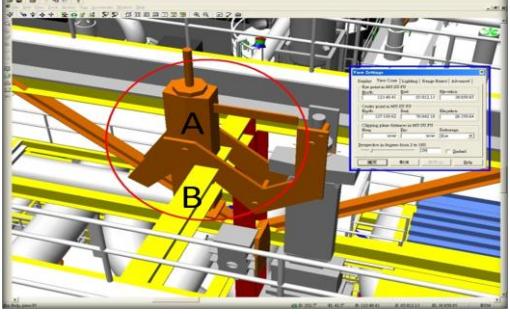
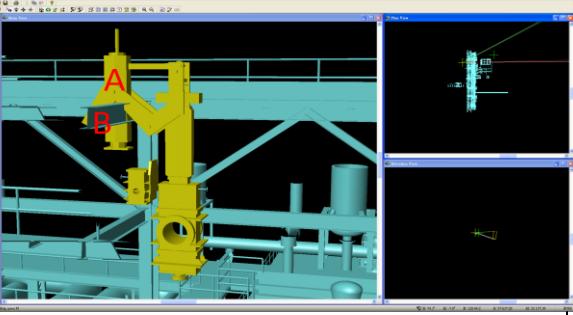
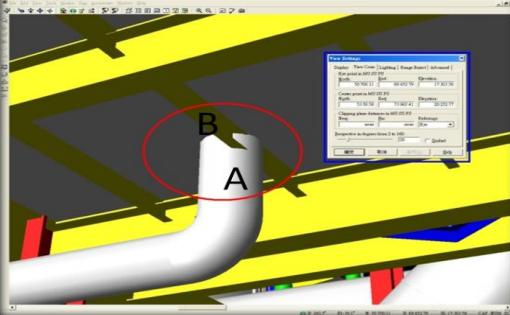
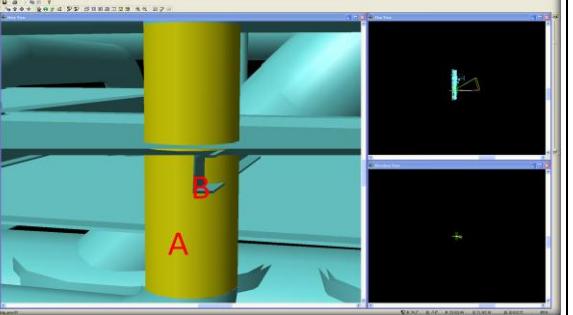
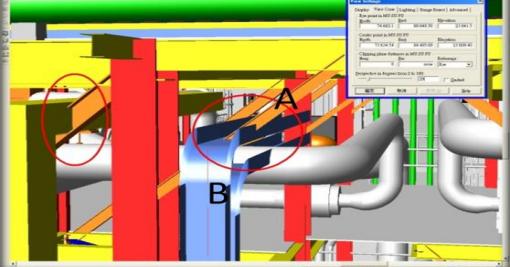
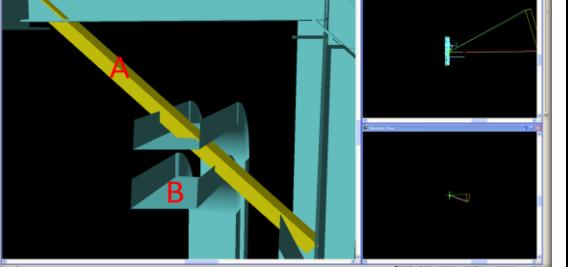
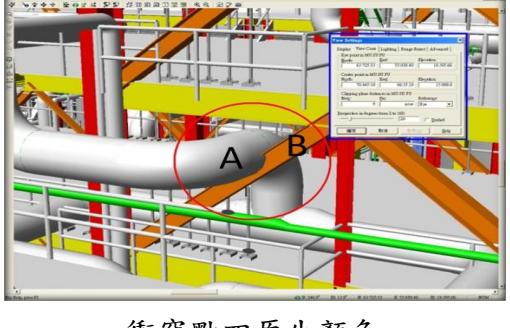
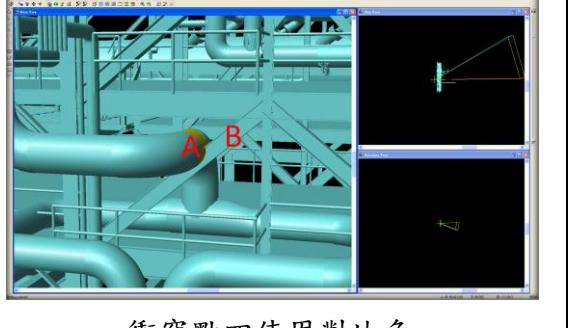


圖 5-6 案例模型對比顏色

表 5-4 案例衝突點對比顏色比較

原來模型衝突點	突顯後衝突點
	
衝突點一原生顏色	衝突點一使用對比色
	
衝突點二原生顏色	衝突點二使用對比色
	
衝突點三原生顏色	衝突點三使用對比色
	
衝突點四原生顏色	衝突點四使用對比色

5.3.3操作情境

開始有 BIM 模型後，由於來自不同設計部門，故需要檢驗是否有衝突，而設計是否有符合需求(需求審查)，若有衝突或需要討論之部分與討論的議題寫入資料庫，然後尋找相關設計單位討論，整個流程如圖 5-7 所示。當相關單位於電腦前準備審查時，首先需確認是否與會者都已連線，如同會議時會確認出席名單，然後與會者登入資料庫中，選擇衝突或需要討論之議題，並且利用本研究之程式同步與會者畫面，但是各個電腦可自由操作自己之模型，如關閉自己不想關心之圖層資料等等，亦不影響其他電腦，也不會影響會議進行，然後進行討論，重複此流程至討論完所有議題，整個流程圖如圖 5-8 所示，以下將以一個較符合實際狀況之情境，沿用本章所使用之範例模型展現整個流程。

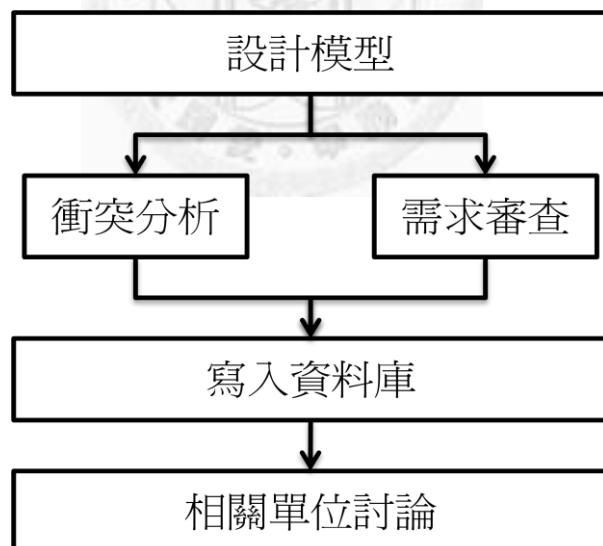


圖 5-7 流程情境

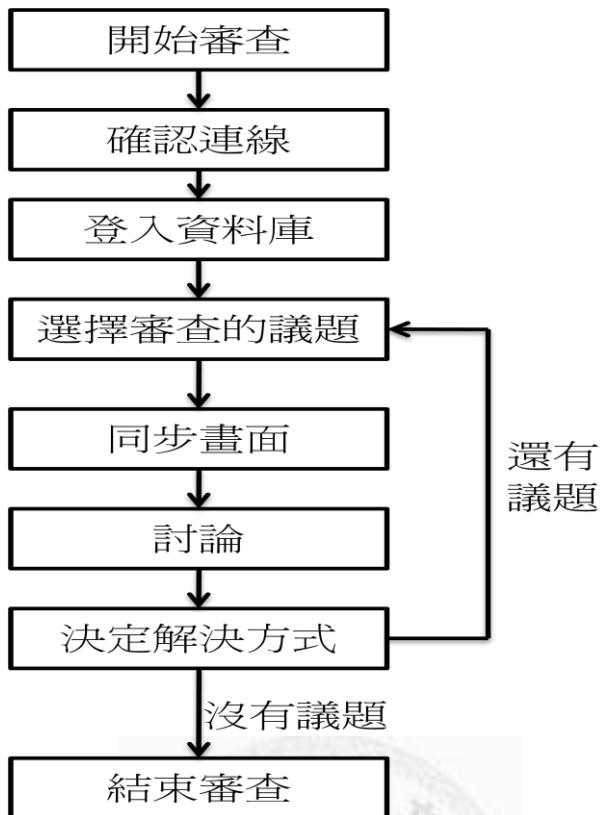


圖 5-8 審查流程情境

一家營造廠 A 承辦一件石化廠之工程，由於石化廠之施工難度很高，因此 A 营造廠打算採用 BIM 之方式設計，利用 3D 模型來設計整個空間位置，於是 A 营造廠之結構部、機電部門與管線部門分別建立 3D 模型，三個部門完成後將 3D 檔案放至一起展示如圖 5-2 所示，三個部門皆認為無任何問題，即將完成檔案交給營造廠 A 之專案管理員，管理員收到檔案後，先執行程式碰撞偵測，發現 20 處部門與部門間碰撞問題並將問題記錄於伺服器中，可供各個部門上網確認問題，並且依據公司處理碰撞之原則如表 5-3 所示解決，最後剩下四個重大衝突如圖 5-3 所示需各部門討論，但由於某部門設計者剛好出國出差，無法面對面討論，剛好那位設計者所處之地有網路，可利用本研究之程式進行討論，便約定時間做討論。

到討論時間，三個部門之設計人員陸續登陸伺服器中，加入討論團隊，亦利用了電話會議讓與會者皆聽得見其他人之發言，會議開始

依序就四個重大的碰撞衝突作討論，首先對於圖 5-3(A)所示監測器之位置作討論，於是選擇伺服器中此筆碰撞資料，然後開啟與會者的 BIM 模型，並且將衝突標顯出來如圖 5-9 所示。

設備部門提及：關於此設備與結構之碰撞，於部門內討論後，認為鋼結構梁 B 於設計上不需要存在，並且在設計時結構部給予資料不完整，導致將重要設備設計於此，若欲更改設計，將牽連附近相關之設備位置，故本部門希望能與結構部討論拿掉鋼梁 B。

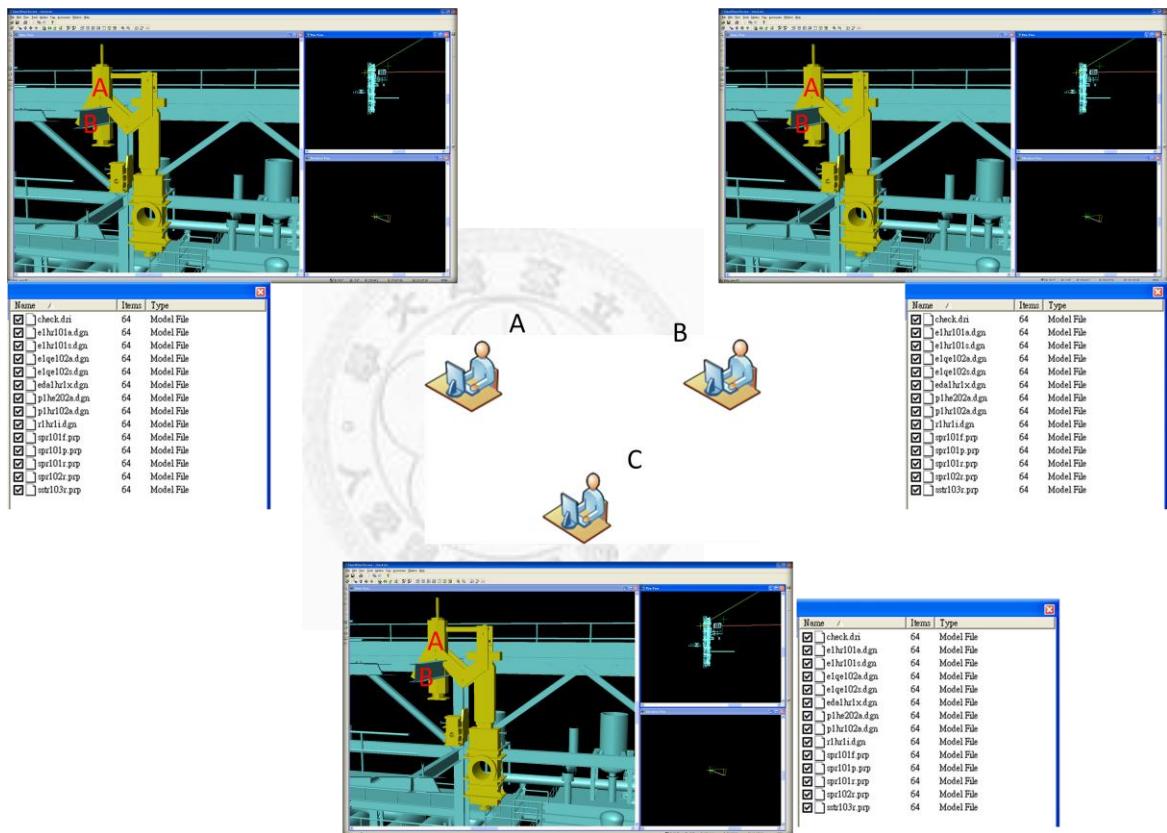


圖 5-9 同步畫面

當設備部門說明同時，結構部門與管線部門邊聽解說邊操作 SPR，開啟或關閉所關心的圖層，並且轉換視角以便他們更了解此衝突點資訊，如圖 5-10 所示。

結構部門提及：關於此衝突點，於剛才設備部門說明時，審慎觀察，對於鋼梁 B 我亦認為是不需，故 2 天內會將確定鋼梁 B 是否要移除。

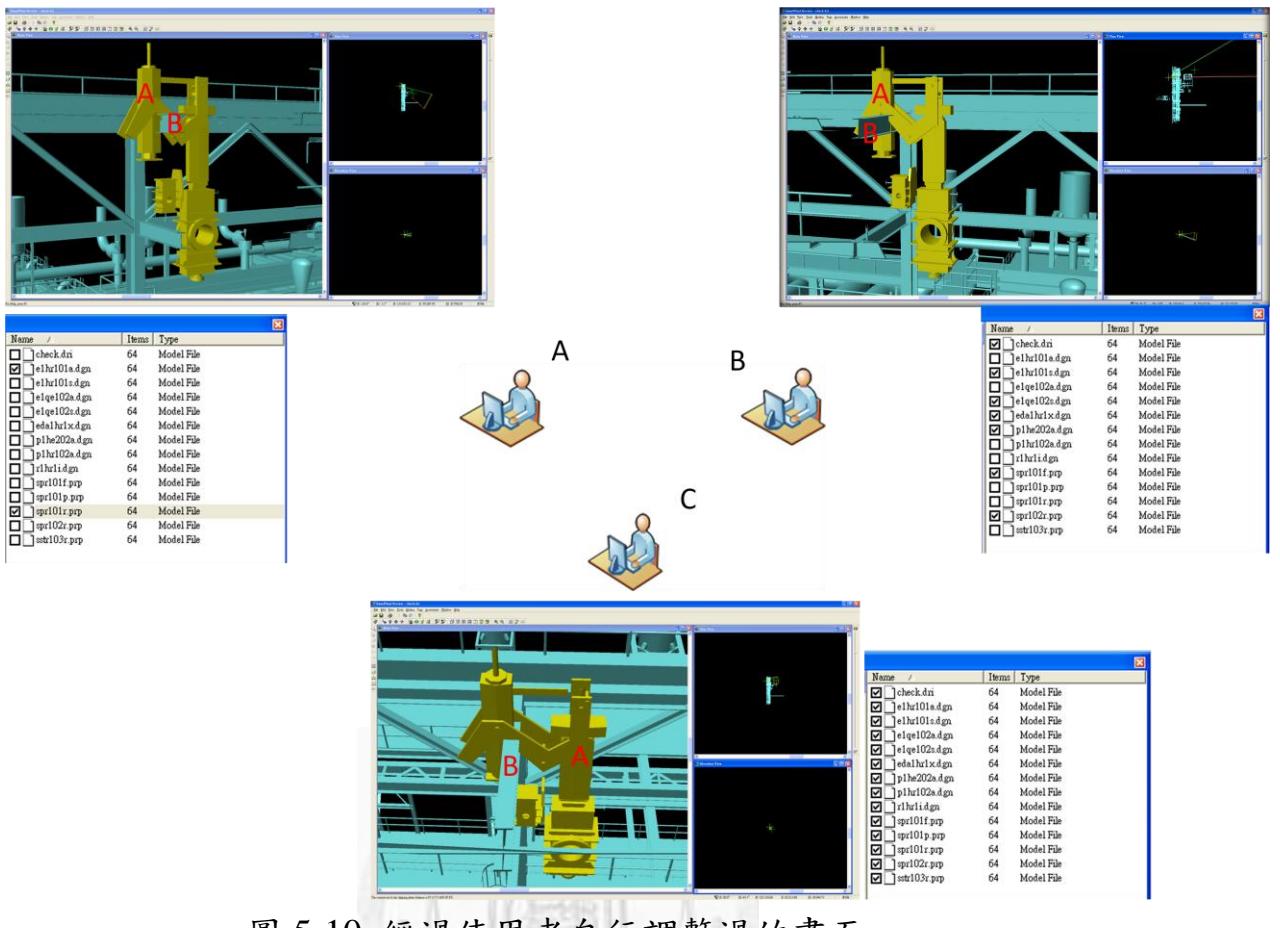


圖 5-10 經過使用者自行調整過的畫面

會議依據流程進行，直到四個議題皆討論完後結束會議。

第六章 結論與建議

6.1. 結論

本研究為應用網路技術與 BIM 軟體，建立網路協同審查作業之架構，使協同審查人員身處異地也能即時同步 3D 資訊，並藉由 3D 協同操作與 BIM 資料整合，支援審查人員或不同合作廠商，解決空間衝突與整合工程介面。

雖然市面上有遠端桌面與虛擬空間的方式，可以達到協同作業的目的，但是對於建築業而言功能不夠全面，遠端桌面方式無法同時給與會者自由操控的空間，其原因在於營建業各團隊所著重的目標皆有所不同，例如結構部門較關心機電設備，但是管線作業對該部門卻不那麼重要，若是使用遠端桌面方式，若是要讓各部門滿足所關心之項目，則需耗費長時間於輪流觀看之流程中，由於隨著會議的進行隨時都有新的會議重點產生，因此與會過程漫長且繁複，反之若與會時間較短則無法滿足各個部門關心項目，造成部門間溝通不良。而虛擬空間方式對於建築業來說，需要更多努力在統一建築資料與整合各家軟體上，因營建專案牽涉相當多團隊，各團隊依據自己所負責工作之特性，選擇特定的或其專業軟體，因此在專案執行面上各團隊擁有自己的獨特模型，在與他人做整合時常常會因所選軟體系列不同，造成模型整合上不易，也因此討論時更加產生介面溝通不良之問題。

本研究提出 BIM.RA 工具，協助協同設計審查之工作，BIM.RA 利用代理者架構，可以達到不同團隊於同時不同地審查，不同軟體藉由 BIM.RA 幫助，達到協同設計審查之效果。BIM.RA 具有複合式用戶端之特性，能讓會者具良好自由度於自己的電腦，並且操控 3D 模型程式之功能，並不為影響其他與會人。

以下為本研究之成果與貢獻：

- 設計具擴充性與適用性之 BIM 代理者架構解決不同軟體協同：
本研究利用以代理者機制為基礎之架構，設計了 BIM 的代理者架構，除了可結合不同的 BIM 軟體，展示現有 3D 模型之物件細節與定位點線面的空間資訊，並且結合單機與網路平台的使用方式進行運作。由於代理者架構為一個平台概念，且利用 SmartPlant Review(SPR) API 開發程式工具驗證了平台之可行性。程式架構本身並不限定用任何語言或 BIM 軟體實作，只要語言本身具有物件導向且能與網路資源做連結；3D 軟體本身之 API 能夠組合出 3.3 介紹之相同功能 API，即可應用於此架構中。架構使用的同時，亦可使用該 BIM 軟體所具有之各項功能，如使用滑鼠轉視角或開啟圖層等，加上是直接呼叫商用軟體，可免去維護 BIM 軟體之麻煩。
- 實作 BIM.RA 驗證代理者架構可行性：
BIM.RA 利用代理者技術，方便開發者藉由網路控制 BIM 軟體展示模型，幫助工程師能更快速且更直覺地展示現有 3D 模型物件。在整個協同設計審查流程中，BIM.RA 扮演核心的角色，利用代理者架構，結合不同的 BIM 軟體進行審查，且結合單機與網路平台的使用方式進行 3D 呈現，此種特色皆適用於所有具空間定位需求之軟體。學界在此已有多方面應用，包括改善傳統的設施維護管理只有文字描述的缺點，如裴騰兆 (2010)所提出之設施物業管理系統，即應用到網路 3D 定位之成果。

- 實作 BIM.CS 實現協同作業:

BIM.CS 於伺服器端利用 XML 技術，處理與會人員資料。主要控制當接收到使用者 API 命令，傳送至其他連線使用者之 BIM.RA，藉由 BIM.RA 處理 3D 模型顯示，使所有使用者皆能與操作者關注同一模型細節。

- 藉由 BIM.RA 與 BIM.CS 發展新的協同審查機制:

藉由 BIM.RA 與 BIM.CS 的幫助，提出設計審查時 BIM.RA、BIM.CS 與 BIM 配合使用之流程，能支援審查人員或不同合作廠商，進行衝突解決與介面整合。

- 提供設計 API 之建議以幫助之後發展設計之參考:

藉由 BIM.RA 設計與實作經驗，整理出為了展示之圖台所必須包含之動作(基本 API)，並且根據本研究之需要，利用基本 API 組合高階 API，本研究之 API 之設計，能提供之後發展類似系統功能設計之參考。

6.2. 未來研究建議

本研究現階段仍以目的與需求做分析初，步設計實作一個工具軟體 BIM.RA，驗證架構之可行性，並以案例作為測試案例而進行測試。BIM.RA 除了可用於協同設計審查之外，也可用於需 3D 展示之系統，例如：4D、物業管理、設施管理、協同審查等。然而因使用 SmartPlant Review 之 API 進行平台的驗證，於功能上僅針對審查做相關之 API 設計，故若需其他之應用，則需擴充 API，以符合使用此架構。本研究著重於平台架構，與可能應用之範圍做探討，以下針對本研究所遭遇到之瓶頸及後續可能發展提出建議：

- 加入其他 3D 模型顯示軟體之連結：目前雖然只有進行實作 SmartPlant Review，但是仍有許多市面上 BIM 軟體，若能將其他性質相同之 BIM 軟體進行連結，將有助於各 BIM 軟體間協同應用。
- 相關資料之顯示整合：對於 BIM 之模型來說，目前只展示靜態之模型狀態，故可以將時間軸加入之展示納入考慮，例如 4D。
- 進入實用階段：使程式架構以及協同審查，與實際之協同人員進行實用，使介面功能得以切確符合使用者之需求。

參考文獻

- Abdel-Wahab, H.M., Guan, S.U., & Nievergelt, J. (1988), Shared Workspaces For Group Collaboration : An Experiment Using Internet And UNIX Interprocess Communications, *IEEE Communications Magazine*, 26(11), 10-16.
- Anumba, C.J., Ugwu, O.O., & Newnham, L. (2002), Collaborative Design Of Structures Using Intelligent Agents, *Automation in Construction*, 11(1), 89-103.
- Autodesk (2009), Improving Building Industry Results through Integrated Project Delivery and Building Information Modeling, Autodesk whitepaper.
- Chen, H.M., & Tien, H.C. (2007), Application of Application of Peer-to-Peer Network For Real-Time Online Collaborative Computer-Aided Design, *Computing in Civil Engineering*, 21(2), 112-121.
- Chen, P.H., Cui, L., Wan, C., Yang, Q., Ting, S.K., Robert, L., & Tiong, R.L.K. (2005), Implementation of IFC-based Web Server for Collaborative Building Design Between Architects and Structural Engineers, *Automation in Construction*, 14(1), 115-128.
- Cohen, P.R., Cheyer, A. Wang, M., & Baeg, S. C. (1994), An Open Agent Architecture. *Autonomous Agents and Multi-Agent Systems*, 4(1-2), 143-148.
- Eastman, C., Teicholz, P., Sacks, R., & Liston, D. (2008), *BIM Handbook: A Guide To Building information Modeling For Owner · Managers · Designers · Engineer And Contractors*, JERRY LAISERIN.
- Froese, T.M. (2005), Information Management for Construction, *Proceedings of the 4th International workshop on Construction Information Technology in Education*, 7-17.
- Fruchter, R. (1996), Conceptual collaborative building design through shared graphics, *IEEE Expert*, 11(3), 33–41.

Fruchter, R. (1999), A/E/C TEAMWORK: A COLLABORATIVE DESIGN AND LEARNING SPACE, *Journal of Computing in Civil Engineering*, 13(4), 261–269.

Gallaher, M. P., O'Connor, A.C., Dettbarn, J.L., Jr., & Gilday, L. T. (2004), *Cost Analysis of Inadequate Interoperability in the U.S. Capital Facilities Industry*, National Institute of Standards and Technology.

Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1995), *Design Patterns Elements Of Reusable Object-Oriented Software*, AddisonWesley Professional.

Gamma, E., Helm, R., Johnson, R., Vlissides, J.著，葉秉哲譯 (2001)，*物件導向設計模式*，培生教育出版集團。

Hsieh , S.H., Chen, C.S., Liao, Y.F., Yang, C.T., & Wu, I.C. (2006), Construction Director: 4D Simulation System For Plant Construction, *Proceedings of the Tenth East Asia-Pacific Conference on Structural Engineering and Construction* , 135-140.

Intergraph, <http://www.intergraph.com/products/ppm/smartplant/review/default.aspx>, 2010.2

Mallon, J. C., Mulligan, D. F. (1993), Quality Function Deployment-A System For Meeting Customer's Needs, *Journal of Construction Engineering and Management*, 119(3), 516-530.

Martin, D. L., Cheyer, A.J., & Moran, D. B. (1999), The Open Agent Architecture : A framework For Building Distributed Software Systems, *Applied Artificial Intelligence*, 13(1), 91-128.

Nwana, H.S. (1996), Software Agent: An Overview, *Knowledge Engineering Review*, 11(3), 1-40.

Owolabi, A., Anumba, C.J., El-Hamalawi, A., & Harper, C. (2006), Development Of an Industry Foundation Classes Assembly Viewer, *Computing in Civil Engineering*, 20(2), 121-131

Shen, W., Han, Q., Mak, H., Nellamkavil, J., Xie, Helen, Dickinson, J., Thomas, R., Pardasani, A., & Xue, H. (2010), Systems Integration and Collaboration in Architecture, Engineering, Construction, and Facilities Management : A Review, *Advanced Engineering Informatics*, 24, 196-207.

The Open Agent Architecture, <http://www.ai.sri.com/~oaa/>, 2010.4

Kvan, T. (2000), Collaborative design : What Is It?, *Automation in construction*, 9(4), 409-415.

Vanlande, R., Nicolle, C., & Cruz, C. (2008), IFC and Building Life Cycle Management, *Automation in Construction*, 18(1), 70-78.

World Wide Web Consortium, <http://www.w3.org/standards/xml/>, 2010.4

市田嵩-牧野鐵治(1998)，設計審查，先鋒企業管理發展中心設計審查小組譯。

呂英瑞(2004)，J2EE 平台與.NET 平台互通方法之實作與比較，國立中央大學資訊工程研究所碩士論文。

吳崇弘(1997)，工程圖檔關聯建構之物件導向分析與設計，國立台灣大學土木工程學研究所碩士論文。

吳翌禎(2007)，多維度工程專案資訊整合管理與視覺化之研究，國立台灣大學土木研究所博士論文。

李勝朗(1998)，工程設計審查方法-以程序系統工程設計為例，工程月刊，第七十一卷第3期。

卓正倫(2005)，設計完成度審查模式之建構-以建築工程為例，國立交通大學土木工程學系碩士論文。

康仕仲、蔡孟涵 (2009)，建築資訊模型之技術發展過程，營建知訊，第 316 期。

張清靠(1983)，設計審查的方法與實務，機械工業月刊，第 47 期。

陳瑞鈴等(2001)，台灣建築生命週期使用年限調查之研究，內政部建築研究所報告。

裴謄兆(2010)，應用建築資訊空間系統與專案導向企業資源管理技術於建築設施維護管理，國立台灣大學土木研究所碩士論文。

蕭樂同(2004)，營區設施維護管理策略之研究，國立中央大學土木研究所碩士論文。



附錄 A: BIM.RA 基本 API

API 名稱	LoadBim(String Filename)
回傳值	當成功會傳 True，失敗回傳 False。
參數	String Filename:代表想要開啟的檔案名稱。
應用	可以讓代理者開啟一個 BIM 檔案。

API 名稱	CloseBim();
回傳值	當成功會傳 True，失敗回傳 False。
參數	沒有。
應用	可以讓代理者關閉 BIM 軟體。

API 名稱	DisplayDefine(String name, String color);
回傳值	當成功會傳 True，失敗回傳 False。
參數	String name:想創造圖層的名稱。 String color:圖層的物件顏色。
應用	定義一個圖層並且給定名字與使用的顏色。

API 名稱	DisplayDelete (String name);
回傳值	當成功會傳 True，失敗回傳 False。
參數	String name:想刪除圖層的名稱。
應用	刪除圖層。

API 名稱	DisplaySet (String name , String Show);
回傳值	當成功會傳 True ，失敗回傳 False 。
參數	String name: 圖層的名稱。 String Show: 設定是否顯示，true 顯示，false 不顯示。
應用	圖層中設定顯示或者不顯示。

API 名稱	RangeFit();
回傳值	當成功會傳 True ，失敗回傳 False 。
參數	沒有。
應用	視角設成剛好可看見所有顯示物件。

API 名稱	ViewUpdate();
回傳值	當成功會傳 True ，失敗回傳 False 。
參數	沒有。
應用	更新畫面。

附錄 B: BIM.RA 高階 API

API 名稱	HighLightOneItem(String Filename, String Linkages)
回傳值	當成功會傳 True，失敗回傳 False。
參數	<p>String Filename:物件所屬在檔案的名字</p> <p>String Linkages:物件在 BIM 軟體中，代表的編號。</p>
應用	對於一個物件顯示特殊顏色讓使用者快速了解在 3D 模型中的點線面的位置與狀態。

API 名稱	HighLightSystemItems(String Filename, String[] Linkages)
回傳值	當成功會傳 True，失敗回傳 False。
參數	<p>String Filename:物件所屬在檔案的名字</p> <p>String[] Linkages:許多物件在 BIM 軟體中，代表的編號，寫成陣列的格式。</p>
應用	對於許多物件顯示特殊顏色讓使用者快速了解在 3D 模型中的點線面的位置與狀態。此種效果可以展現系統的概念在圖層之中。

API 名稱	HighLightRangeFitOneItem(String Filename,String Linkages);
回傳值	當成功會傳 True，失敗回傳 False。
參數	String Filename:物件所屬在檔案的名字 String Linkages: 物件在 BIM 軟體中，代表的編號。
應用	對於一個物件顯示特殊顏色並且將視角符合所關注的物件大小。

API 名稱	HighLightRangeFitSystemItems(String Filename,String Linkages[]);
回傳值	當成功會傳 True，失敗回傳 False。
參數	String Filename:物件所屬在檔案的名字 String[] Linkages:許多物件在 BIM 軟體中，代表的編號，寫成陣列的格式。
應用	對於很多物件顯示特殊顏色並且將視角符合所關注的所有物件包含的視角大小

API 名稱	HighLightOnlyOneItem(String Filename,String Linkages);
回傳值	當成功會傳 True，失敗回傳 False。
參數	String Filename:物件所屬在檔案的名字 String Linkages: 物件在 BIM 軟體中，代表的編號。
應用	對於一個物件顯示特殊顏色其他不相關的物件隱藏起來 讓使用者快速了解在哪裡。

API 名稱	HighLightOnlySystemItems(String Filename,String[] Linkages);
回傳值	當成功會傳 True，失敗回傳 False。
參數	String Filename:物件所屬在檔案的名字 String[] Linkages:許多物件在 BIM 軟體中，代表的編號， 寫成陣列的格式。
應用	對於很多物件顯示特殊顏色其他不相關的物件隱藏起來 讓使用者快速了解在哪裡。

API 名稱	HighLightRangeFitOnlyOneItem(String Filename,String Linkages);
回傳值	當成功會傳 True，失敗回傳 False。
參數	String Filename:物件所屬在檔案的名字 String Linkages: 物件在 BIM 軟體中，代表的編號。
應用	對於一個物件顯示特殊顏色其他不相關的物件隱藏起來 並且將視角符合所關注的物件大小。

API 名稱	HighLightRangeFitOnlySystemItems(String Filename,String Linkages[]);
回傳值	當成功會傳 True，失敗回傳 False。
參數	<p>String Filename:物件所屬在檔案的名字</p> <p>String[] Linkages:許多物件在 BIM 軟體中，代表的編號，寫成陣列的格式。</p>
應用	<p>對於很多物件顯示特殊顏色其他不相關的物件隱藏起來</p> <p>並且將視角符合所關注所有物件包含的視角大小。</p>



附錄 C:Server 與 BIM.CS API

API 名稱	broadcastAction(String Action, String Adress);
回傳值	當成功會傳 True，失敗回傳 False。
參數	String Action:使用者操作的動作名稱。 String Adress:想傳送到的 IP 位址。
應用	把動作傳送到指定的 IP 位址，通知協調代理者動作。

API 名稱	Join(String name, String Online, String Adress);
回傳值	當成功會傳 True，失敗回傳 False。
參數	String name:欲參與者的名字。 String Online:是否在線上。 String Adress:欲參與者的 IP 位址。
應用	把欲參與者的資訊寫入伺服器端的 XML，檔案格式如圖 3-4 所示。

API 名稱	Leave(String name):
回傳值	當成功會傳 True，失敗回傳 False。
參數	String name:想從 XML 檔案格式中刪除與會者的名字。
應用	從 XML 檔案中找到名字，並且刪除相關的資料。

API 名稱	XMLreader();
回傳值	當成功會傳 True，失敗回傳 False。
參數	沒有
應用	讀伺服器端的 XML 檔案，並且記錄到程式當中。

API 名稱	XMLwriter();
回傳值	當成功會傳 True，失敗回傳 False。
參數	沒有
應用	將程式中的與會人資料，寫入 XML 檔案。

API 名稱	clearAllPerson();
回傳值	當成功會傳 True，失敗回傳 False。
參數	沒有
應用	刪除所有在 XML 中的資料。

API 名稱	GetPerson();
回傳值	當成功會傳 True，失敗回傳 False。
參數	沒有
應用	獲得 XMLreader()後的程式中與會者的資料。

作者簡歷



姓名:何松柏

出生地:彰化縣

生日:74 年 12 月 21 日

2008~2010 國立台灣大學土木工程學研究所電腦輔助工程組

2004~2008 國立中興大學土木工程學系

2001~2004 台中高級私立衛道中學